HELSINKI UNIVERSITY OF TECHNOLOGY

Department of Electrical and Communications Engineering

Laboratory of Telecommunications Technology

Jukka Halonen

Service Activation in IP networks – requirements for provisioning mediation system

Thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Engineering

Helsinki, September 31, 2001

Supervisor: Professor Jorma Jormakka

Instructor: Jarkko Huuhtanen, M.Sc.

TKK Sähkö- ja tietoliikennetekniikan kirjasto
Otakaari 5 A
02150 ESPOO
07 -11- 2001

Author:	Jukka Halonen
Name of the Thesis:	Service activation in IP networks – requirements for provisioning mediation system
Date: September 31, 2001	Number of pages: 89 and 3 Appendixes

Department of Electrical and Communications Engineering:

Professorship: S-38 Telecommunications Technology

Supervisor: Professor Jorma Jormakka

Instructor: M.Sc Jarkko Huuhtanen

Virtual private network is one of the fundamental services that service providers must be able offer to their customers. There are many technologies for implementing virtual private networks, thus there are many ways to activate them. Service providers must be able to activate these services quickly and efficiently by using tools that enable automated network-wide service activation.

Comptel is the market leader in mediation devices for telecommunication networks. However, the convergence of internet protocol and telecommunication worlds has led to a situation where Comptel must be able to offer mediation device solutions also to internet protocol world. One of the goals of this study is to study the service activation requirements and different provisioning interfaces that virtual private networks have.

One of these interfaces is lightweight directory access protocol. It is widely used in several hardware and software vendors' products. This thesis introduces a solution that enables Comptel's provisioning mediation system to interface directories with lightweight directory access protocol interface. This solution is a generic network element interface module that uses aggressive run-time parameterisation.

Keywords: Virtual private network, MPLS, LDAP, MDS/SAS

Tekijä: Jukka Halonen

Työn Nimi: Service Activation in IP networks – requirements for provisioning mediation system

Päivämäärä: Syyskuun 31.,2001 Sivumäärä: 89 ja 3 liitettä

Osasto: Sähkö ja Tietoliikennetekniikan osasto

Professuuri: S-38 Teletekniikka

Työn valvoja: Professori Jorma Jormakka

Työn ohjaaja: DI Jarkko Huuhtanen

Virtuaalinen yksityisverkko on yksi tärkeimmistä palveluista, jota palveluntarjoajat haluavat myydä asiakkailleen. Virtuaalinen yksityisverkko voidaan toteuttaa monella eri tavalla, käyttäen erilaisia teknologioita. Tämän takia palvelu voidaan myös aktivoida monella eri tavalla. Palveluntarjoajille on tärkeää, että he voivat aktivoida palvelun nopeasti ja kustannustehokkaasti. Tätä varten he tarvitsevat työkaluja, joilla palvelunaktivointi voidaan automatisoida.

Comptel on markkinajohtaja telekommunikaatioverkkojen mediaattorijärjestelmien toimittajana. Internetin ja telekommunikaatioverkkojen konvergenssi on johtanut tilanteeseen, jossa Comptelin täytyy pystyä toimittamaan mediaattori järjestelmiä myös internetverkkoihin. Yksi tämän työn tarkoitus onkin tutkia mitä vaatimuksia palvelun aktivoinnilla on virtuaalisissa yksityisverkkoissa.

Yksi näistä rajapinnoista on lightweight directory access protocol. Tämä työ esittelee ratkaisun kyseiselle rajapinnalle. Ratkaisu perustuu generiseen ohjelmisto moduuliin, joka mahdollistaa Comptelin mediaattoriohjelmiston yhdistämisen hakemistoihin, jotka käyttävät LDAP rajapintaa.

Avainsanat: Virtual private network, MPLS, LDAP, MDS/SAS

Acknowledgements

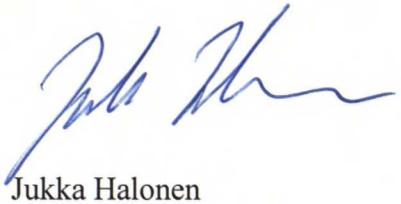
This thesis has been done in Comptel service provisioning unit in Helsinki. The thesis has been instructed by M.Sc. Jarkko Huuhtanen. I would like to thank Jarkko for guiding me on my work. I would also like to thank Comptel for giving me the opportunity to do this thesis.

Professor Jorma Jormakka has been the supervisor of my thesis. I would like to thank Jorma for his interest towards my work.

Especially I would like to thank my family for giving me so much support during my studies. I would never have managed to graduate without their help.

Finally I would like to thank my beloved fiancée Sanna for bearing up with me during my thesis. I know that I was not the easiest boyfriend around.

Helsinki 31.9.2001



Jukka Halonen

Table of Contents

Table of Figures

Abbreviations

AAA	Access, Authentication, Accounting
ABR	Available Bit Rate
ACL	Access Control List
AL	Access List
ADSL	Asynchronous Digital Subscriber Line
AH	IP Authentication Header
ANSI	American National Standard Institute
API	Application Programming Interface
AS	Autonomous Systems
ATM	Asynchronous Transfer Mode
ATMP	Ascend Tunnel Management Protocol
BGP	Border Gateway Protocol
BRI	Basic Rate Interface
CBR	Constant Bit Rate
CCS	Customer Care System
CCITT	Consultative Committee on International Telephone and Telegraph
CE	Customer Edge
CEF	Cisco Express Forwarding
CHAP	Challenge Handshake Authentication Protocol
CIM	Common Information Model
CLI	Command-Line Interface
CPE	Customer Premises Equipment
CSS	Content Service Switch
DEN	Directory Enabled Networks
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiated Services
DIT	Directory Information Tree
DLCI	Data Link Connection Identifier
DMTF	Desktop Management Task Force
DN	Distinguished Name
DNS	Domain Name Server

DoS	Denial of Service
DSCP	Differentiated Services Code Point
DTE	Data Terminal Equipment
EGP	Exterior Gateway Protocol
ESP	Encapsulating Security Payload
ER	Edge Router
FEC	Forwarding Equivalence Class
FR	Frame Relay
FTP	File Transfer Protocol
GFR	Guaranteed Frame Rate
GPRS	General Packet Radio System
GRC	Global Resource Configuration
GRE	Generic Routing Encapsulation
GSM	Global System for Mobile communication
GUI	Graphical User Interface
HLR	Home Location Register
HQ	Headquarters
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IKE	Internet Key Exchange
IP	Internet Protocol
IPsec	Internet Protocol security
ISAKMP	Internet Security Association and Key Management Protocol
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
ITU-T	International Telecommunications Union – Telecommunication Sector
L2F	Layer 2 Forwarding
L2TP	Layer 2 Tunnelling Protocol
LAC	L2TP Access Concentrator
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LDIF	LDAP Interchange Format
LDP	Label Distribution Protocol

LER	Label Edge Router
LEX	Local Command Executor
LIB	Label Information Base
LL	Leased Line
LNS	L2TP Network Server
LSP	Label Switched Path
LSR	Label Switched Router
MAC	Media Access Control
MAN	Metropolitan Area Network
MDS	Mediation Device Solution
MIB	Management Information Base
MPLS	Multiprotocol Label Switching
MSC	Mobile Switching Centre
NAS	Network Access Server
NE	Network Element
NMS	Network Management System
nrt-VBR	non-real-time Variable Bit Rate
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
P	Provider
PAP	Password Authentication Protection
PBR	Policy Based Routing
PDU	Protocol Data Unit
PE	Provider Edge
PN	Private Network
PoS	Packet-over-Sonet/SDH
POTS	Plain Old Telephony Service
PP	Point-to-point
PPP	Point-to-point Protocol
PPTP	Point-to-Point Tunnelling Protocol
PRI	Primary Rate Interface
PSN	Packet-Switched Network
PSTN	Public Switched Telephone Network

PVC	Permanent Virtual Circuit
QoS	Quality of Service
RADIUS	Remote Authentication Dial-In User Service
R&D	Research and Development
RFC	Internet Request for Comments
RD	Route Distinguisher
RDN	Relative Distinguished Name
RR	Route Reflector
RSVP	Resource ReSerVation Protocol
rt-VBR	real-time Variable Bit Rate
SA	Security Association
SAS	Subscriber Administration System
SASL	Simple Authentication and Security Layer
SDH	Synchronous Data Hierarchy
SDK	Software Development Kit
SLA	Service Level Agreement
SMP	Service Management Point
SNMP	Simple Network Management Protocol
SOHO	Small Office / Home Office
Sonet	Synchronous Optical Network
SVC	Switched Virtual Circuit
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UBR	Unspecified Bit Rate
UI	User Interface
UMTS	Universal Mobile Telephony System
UDP	User Datagram Protocol
UMTS	Universal Mobile Telephone System
URL	Uniform Resource Locator
VC	Virtual Circuit
VCI	Virtual Circuit Identifier
VMS	Voice Mail System
VoD	Video-on-Demand

VoIP	Voice-over-IP
VP	Virtual Path
VPI	Virtual Path Identifier
VPN	Virtual Private Network
VPNSC	VPN Service Center
VRF	VPN routing/forwarding instance
WAN	Wide Area Network

1 Introduction

Service activation has been understood in the telecommunication world as transfer of subscriber information from operator's Customer Care System (CCS) into operator's network elements [Huuh99]. Comptel's provisioning mediation system called Mediation Device Solutions - Subscriber Administration System (hereafter referred to as MDS/SAS) has been successfully used in several telecommunication operators' networks for service activation. It has been widely deployed in different types on networks, such as Global System for Mobile communication (GSM), Plain Old Telephony Service (POTS) and satellite telephone networks. One common denominator with all these networks is that they are based on centralised and quite complicated Network Elements (NEs).

New emerging technologies, such as General Packet Radio System (GPRS) and Universal Mobile Telephony System (UMTS), are accelerating the convergence of Internet Protocol (IP) and telecommunication worlds. This has led to a situation where telecommunication operators require mediation devices that are able to operate in both worlds. Service providers are looking for overall solutions that enable automated service activation without large administrational overhead. Thus, operators require that MDS/SAS is able to activate services also in the IP network elements.

Also the pure IP based networks offer many attractive possibilities for Comptel. There are lots of IP services that require subscriber based service activation. Virtual Private Network (VPN) is one of the most interesting IP services that service providers are offering to their customers. Service providers have similar needs to active these services fast and economically as telecommunication operators have with their networks. However, the requirements for service activation are usually quite different between these networks. The main difference between telecommunication network and IP network is that in IP network the mediation device should also be able to perform some network management actions and not just to be able to transfer subscriber information into the network elements.

MDS/SAS has already been used in several IP based implementations. However, there are many services that MDS/SAS has not been tested with and a more comprehensive view of the requirements that service activation in IP networks have is needed. The ideal situation would be that MDS/SAS is the ultimate solution for this problem. One of the goals of this study is to gain experience of the different technologies and network elements that MDS/SAS should be able to operate with.

1.1 Research Objectives

The goal of this study is to study different IP-VPN solutions in order to identify the provisioning requirements that these solutions introduce. Concrete provisioning requirements are derived from a Multiprotocol Label Switching (MPLS) based VPN solution. These requirements lay the foundation for the second part of the study. The second part of the study is to implement one of the new network element interfaces that VPN solutions have. Lightweight Directory Access Protocol (LDAP) is chosen as the network element interface that will be implemented in this study. The provisioning requirements, for the LDAP interface module, are derived from the example MPLS based VPN network.

1.2 Research Methods

The first part of the thesis is a literature study, which is based on technical literature and papers, standards, Internet homepages and product brochures. This part gives an overall description of MDS/SAS. It also studies the different technologies that are used to implement VPNs. Additional emphasis is given to MPLS based VPN solution since it is one of the most important technologies that customers have asked Comptel to support. The MPLS based VPN solution, which is used to study the actual service activation requirements, is a generic implementation.

The second part will give a detailed description of LDAP. The service activation requirements, that lay the foundation for the implementation part of this study, are

derived both from the example VPN implementation and the general LDAP description. The goal is to design a network element interface that is a generic solution. The final product should be able to operate with several LDAP implementations.

1.3 Structure of the Study

The second chapter gives an overview of MDS/SAS.

The third chapter will study the different VPN solutions. Several technologies and standards are studied in order to get a comprehensive view of the different types of VPN implementations. Also a comparison between the pros and cons of these solutions is done.

The fourth part will describe an example MPLS based VPN solution and the provisioning interfaces within the network. The provision requirements are derived from this chapter.

The fifth chapter gives an introduction to LDAP.

The sixth chapter will concentrate on the network element interface implementation.

The seventh chapter will conclude this thesis.

2 Subscriber Administration System

2.1 Overview

MDS/SAS is a service mediation system through which the telecommunications network operator can manage subscriber information in versatile telecommunication networks. MDS/SAS is used to create subscribers, modify and query subscriber data, activate new services to existing subscribers and delete subscribers and their services in the telecommunications network. MDS/SAS automates the supplying and activating of telecommunications services, so that no manual intervention is needed from the customer care system. The customer care system does not need to have profound knowledge of the telecommunications network itself; MDS/SAS is the interface that takes care of the communication with network elements. Figure 2-1 illustrates the position of MDS/SAS in the telecommunications management network. There can be several types of Network Elements, such as Service Management Point (SMP), Mobile Switching Centre (MSC), and Voice Mail System (VMS). [Comp01]

The customer care system sends requests to and receives responses from MDS/SAS across the application protocol interface. MDS/SAS sends the commands generated from the service requests to the network elements across the network element interface. Also the network elements can send response data to MDS/SAS across this interface. MDS/SAS stores logged data on all of its activities in the database for troubleshooting and monitoring purposes. MDS/SAS can also send error messages to a centralised alarm handling system. MDS/SAS triggers alarms in the case of serious network errors, e.g. if a network element connection goes down. [Korp95]

The MDS/SAS runs typically on background and does not require any specific user interaction. However, the system administrator can monitor the service provisioning activities, manage the MDS/SAS processing and configure new managed network elements into the system through the Graphical User Interface (GUI).

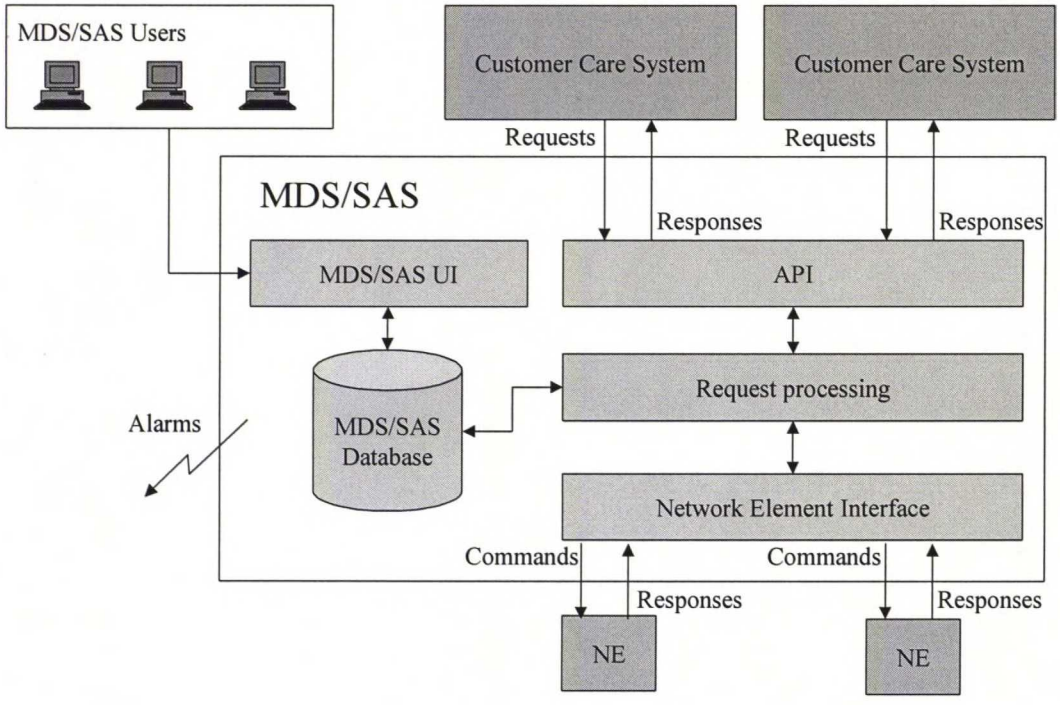


Figure 2-1: MDS/SAS in the Telecommunications Management Network

One of the most important feature MDS/SAS offers to the customer care systems is a standardized, well-defined interface, which enables a transparent network for the CCS. This means that changes made in the network level do not have an impact on the CCS applications. Any changes made in the network, say new version of a HLR, are configured only in the network element layer of MDS/SAS. A telecommunications network typically consists of various types of network elements manufactured by several vendors. Even network element versions from one vendor may have differences. MDS/SAS is a flexible system that is able to support a wide range of different network element types, vendors and software levels.

2.2 Functionality

Service provisioning is handled in a way that MDS/SAS receives work orders from

the customer care system to provision services to subscribers. The term used for the work order is a request. Once MDS/SAS has received a request, it finds the network element(s) where the service needs to be activated in, e.g. in order to create a prepaid Global System for Mobile communication (GSM) subscriber MDS/SAS must add subscriber information to Home Location Register (HLR), VMS and SMP. MDS/SAS translates the request into network element specific commands and then executes these commands. MDS/SAS guarantees that the requests are executed in the defined order and that several requests concerning one subscriber are not executed simultaneously. MDS/SAS prepares a response for each completed request and delivers the response back to the customer care system. The response informs the customer care system how the execution of the service operation has succeeded.

MDS/SAS provides user authentication and authorisation. User authentication is based on a password and user authorisation limits a user's privileges to perform operations within network elements. All information on users' privileges is stored in user profiles. User profile defines which network elements the customer care system's service requests are allowed to address, and what operations (create, modify, display, delete) a CCS is allowed to ask the MDS/SAS system to perform. For example, one customer care system can be allowed only to query subscriber information within a network element, while another may modify, delete and create subscribers within the same network element. Moreover, the user profile can be used to define the privileges that each GUI user has in the system, e.g. which users are allowed to modify the system configurations, manage the user profiles, or just to monitor the system.

In the network layer MDS/SAS translates service-provisioning tasks to network element specific messages and finally sends the messages to the appropriate network elements. In this thesis that network element interface is Lightweight Directory Access Protocol (LDAP). Therefore the network element interface module uses LDAP Application Programming Interface (API) functions to execute requests. MDS/SAS also interprets the responses received from the network elements and stores them in the request log. MDS/SAS sends the response including the result of the task execution to the customer

care system.

2.3 Architecture

The MDS/SAS system is divided into several management layers. Figure 2-2 illustrates the layers of MDS/SAS. [Comp01]

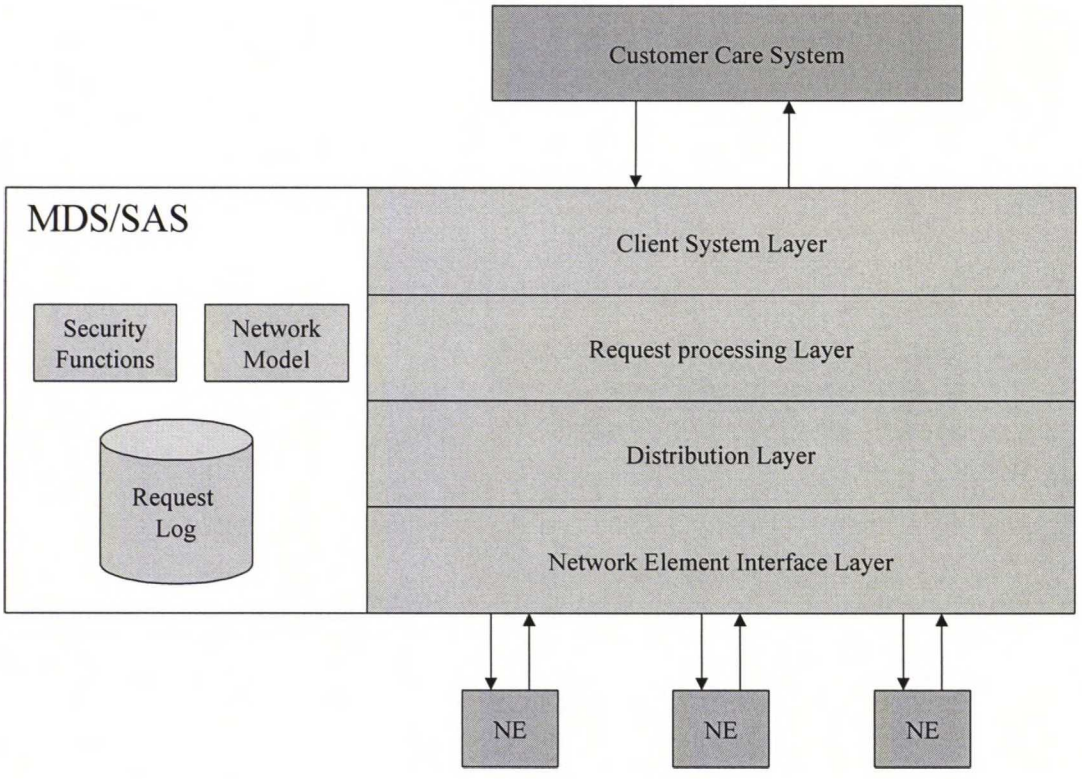


Figure 2-2: MDS/SAS System Layers

The client system layer integrates the customer care system with the MDS/SAS system. In this layer requests are accepted and converted into MDS/SAS internal request format. A part of this translating procedure is checking that the user is allowed to send requests to the MDS/SAS system. MDS/SAS offers several types of interfaces to the CCS including CORBA and text-based.

In the request-processing layer the user's privileges to perform operations within network elements are checked. Necessary conversions are made to a request to generate it into executable tasks. The conversions are made according to the configured rules, such as routing, service packages, number conversions, and supplementary service handling. In this layer it is also ensured that requests concerning one subscriber are run in the order they arrived in the system and that only one request concerning the subscriber is run at a time. Also, once a request has been completed, a response is prepared in this layer to inform the customer care system of how the request execution has succeeded.

In the distribution layer the tasks are distributed to the target network elements for execution. The main role of this layer is to manage the network element specific task queues and ensure that the network resources are used effectively and tasks are executed in the defined order.

The network element interface layer communicates with the network elements. The network element interface module operates on this level. In this layer tasks are translated into network element specific commands, which are then sent to the network elements for execution. Also, this layer receives and interprets the responses from the network elements concerning the task execution.

The MDS/SAS network model describes the telecommunications management network where the system is operating. To be able to control the network elements, MDS/SAS requires knowledge about the underlying telecommunications network. The network model does not belong to any one layer but is "present" throughout the request processing. It is responsible for maintaining the network topology and status information. The network model maintains and updates information on network elements, connections between the network elements and status of the network elements and connections.

3 Virtual Private Network

3.1 Introduction

The definition of VPN is not unambiguous. There are several vendors that offer different types of VPNs based on different technologies. Also the business usage of different VPNs can be totally different. One really cannot find any common denominator for the phase "Virtual Private Network". However, one simple definition for VPN goes like this: *Emulation of a private Wide Area Network (WAN) facility using IP facilities (including the public Internet or private IP backbones)* [RFC2764]. Basically VPN is an interconnection of local area networks (LANs) or/and remote users, making use of a public network, e.g. the Internet. Thus, VPN facilitates the operations of a private network by using interconnections over a public network.

There are three types of VPNs: Access VPNs, Intranet VPNs and Extranet VPNs. Access VPNs connect remote users and Intranet VPNs connect fixed locations such as Small Office / Home Office (SOHO) to a corporate network. Extranet VPNs can be used to give customers and subcontractors a limited access to corporate resources. Intranet and Extranet VPNs are basically the same service, since they are implemented using same technologies. Table 3-1 shows these VPN services and some technologies related to them. [Cisc01]

Table 3-1: VPN Services

Service	Architecture	Technologies
Access VPN	Client Initiated / NAS Initiated	L2TP, IPsec
Intranet & Extranet VPN	IP Tunnel	GRE, IPsec
Intranet & Extranet VPN	Virtual Circuit	Frame Relay, ATM
Intranet & Extranet VPN	MPLS	IP or IP+ATM

General VPN requirements are opaque packet transport, data security and QoS

guarantees [RFC2764]. These requirements are applicable to all the three VPN services mentioned above. However, we must consider these requirements only as guidelines due to the diversity of VPN solutions. The actual implementations of these requirements can be done in many ways or they can be omitted all together. E.g. customers can manage data security by themselves and VPN service provider only provides the VPN data transport service.

Unlike many other technological buzzwords today, such as IP Telephony and UMTS, VPN is already showing its potential. Forrester research's research (Figure 3-1) shows that 73% of the Fortune 1000 companies are moving away from private networks to Virtual Private Networks. Also a good indicator of the potential of VPN is the number of companies that are thinking of using VPN solutions to implement their business-to-business extranet links (Figure 3-1). [Broa98]

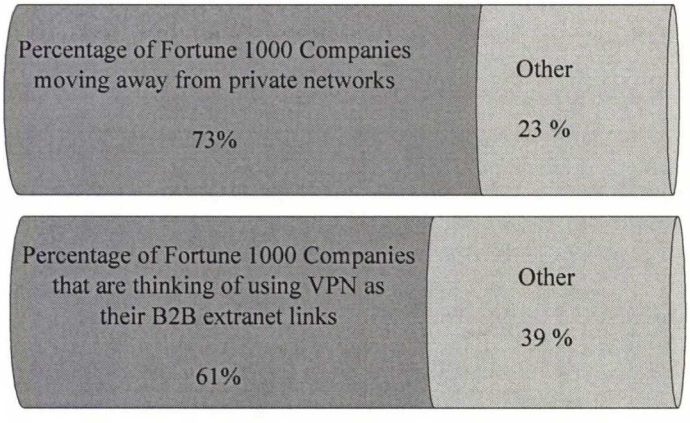


Figure 3-1: VPN market study

This chapter will give the reader an overview of the different types of VPNs and the technologies that are used in implementing them. However, a more detailed description of MPLS based VPN is given since service activation requirements are driven from a

customer case where MPLS technology in used in the core network.

3.2 Drivers for Virtual Private Networks

The utilisation of the Internet has grown tremendously. Many of the today's enterprises (e.g. Amazon) use the Internet as one of their primary communication medium. Even though the Internet is valuable due to its global reach the use of the Internet has severe downsides. One major problem related to the Internet is that it cannot meet the requirements of some of today's IP based services such as security and Quality of Service (QoS). E.g. many eBusiness projects have been delayed due to the fact that the Internet is not able to support the security that these services may require [Data01]. VPN technologies like IP security (IPsec) can enable new IP services such as secure money transaction, secure remote access, etc.

One major driver for VPN is the cost savings opportunity. There has always been a need for private networks for interconnecting remote sites. VPN is thought to be a more economical solution by enabling secure virtual circuits over a public network, thus private networks over the least expensive public data network – the Internet. There are many estimates about the cost savings that a corporate customer may get by moving from leased lines to VPN based Wide Area Networks (WANs). One thing in common to these estimates is that the cost savings can be tremendous. One estimate gives from 50% to 70% cost savings over traditional remote access solutions [Kosi98].

Service Providers are looking for ways to generate more revenues. They have previously concentrated on consumer customers, but have now realised that individual consumers do not create adequate revenue streams. For this reason service providers are turning their focus on corporate customers who are both willing and able to pay for competitive services. Virtual Private Network is a good candidate for such a service. One of the key concepts in VPN is the possibility to easily offer outsourced WAN and Metropolitan Area Network (MAN) solutions to end-users. By using VPNs the corporate customers can outsource their IT investments and concentrate on their core businesses.

International Data Corporation estimates that demand for outsourced network management services will rise to 4,7 billion US$ in 2002 [Broa98].

Private networks that are based on leased lines have major scalability problems. Global companies with many offices and factories dispersed all over the world must be able to interconnect their remote sites together. In order to interconnect all these sites together a mesh topology (Figure 3-2) is required. A full mesh topology is a topology where every party is connected to every other party. Thus, private networks that interconnect various sites can easily become unwieldy and unmanageable. A VPN solution such as MPLS based VPN, enables more scalable and manageable network infrastructure. While VPN is a major replacement threat to traditional large-scale WAN technologies, it is also a good solution for small companies that are building their first wide area network.

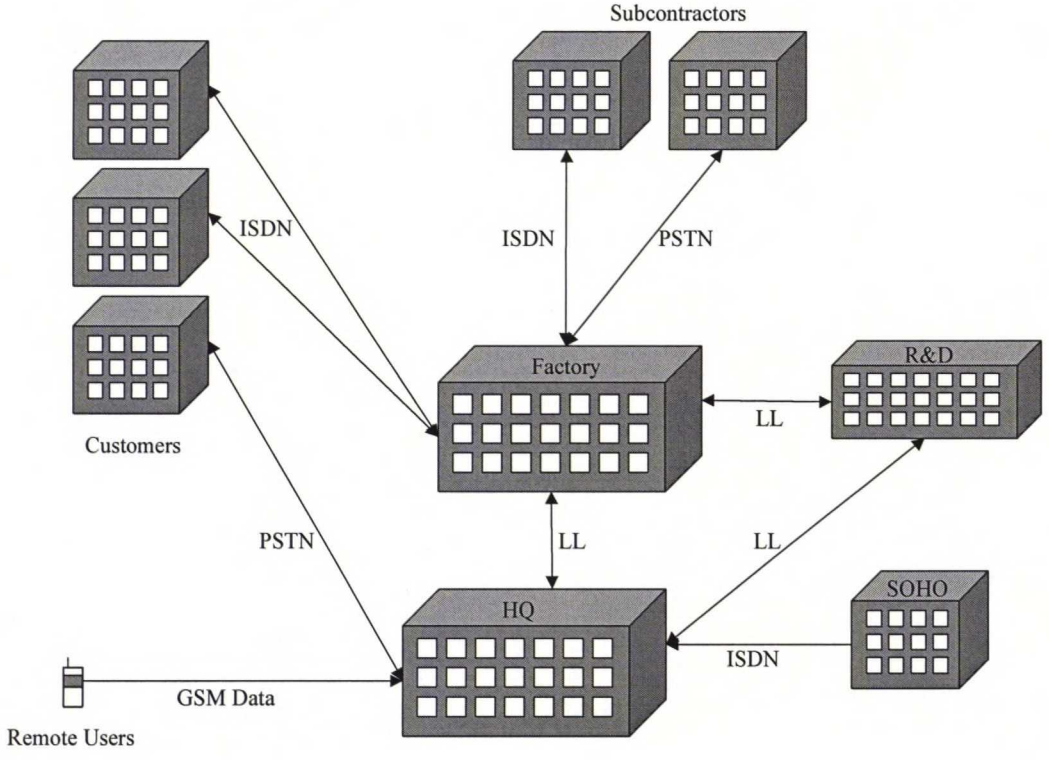


Figure 3-2: Mesh topology

Virtual Private Network seems to benefit both the service provider and the end-user. Service provider can generate new revenue streams from new services and outsourcing. Customer, on the other hand, can reduce IT cost by using the best (and perhaps the cheapest) VPN solution that competing service providers are offering. Thus, many service providers see VPN as one of the fundamental IP services that they must offer in order to attract corporate customers.

3.3 Virtual Private Network Taxonomy

There are many ways to categorise Virtual Private Networks. Technological taxonomy is used due to the fact that it enables logical differentiation of VPNs in the network level (ISO-OSI layers 1 - 3). Technological taxonomy separates the VPNs (and PNs) based on the technologies that are used to implement VPNs. One must remember that this separation does not mean that the technologies/protocols from one category to another are exclusive, quite the contrary. For example MPLS is especially developed to make most of ATM and IP networks: e.g. ability to guarantee the QoS of ATM and the scalability and flexibility of IP.

In figure 3-3 one can see division of VPNs into two principal categories: Overlay and peer-to-peer [Moto00]. The most distinctive mark between these two categories is that the overlay model separates the provider and customer networks so that there is transparency between these networks. The peer-to-peer model, on the other hand, assumes that the customer and provider networks use the same network protocol. Figure 3-3 shows also some rival private network solutions that VPN competes with.

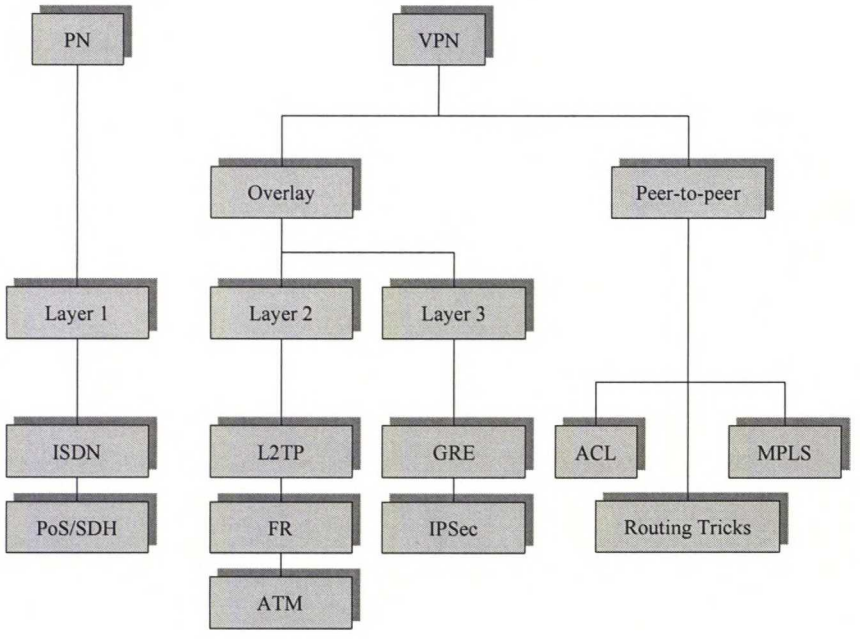


Figure 3-3: VPN Taxonomy

3.3.1 Physical layer (Overlay model)

Leased Line

Leased line (LL) is a permanent connection that is usually leased from a telephone company. Corporate customers have used leased lines for decades to link remote sites together. While this approach usually delivers good connectivity with guaranteed bandwidth, it can offer high bandwidth connections only at fairly high cost. Leased lines are often implemented using Integrated Services Digital Network (ISDN) or Packet over Sonet (PoS).

Packet over Sonet connections has been traditionally implemented using Point-to-Point Protocol (PPP). The problem with this approach is that the current solutions cannot guarantee many of the features that VPN service providers require such as QoS and traffic engineering. There is, however, research work done affiliated to the Abilane project for

accommodating many of the advanced IP facilities to the service architecture. [Pyyh99]

Primary Rate Interface (PRI) ISDN connection can offer up to 2 MB connections for data, voice, fax, video and other source material. ISDN is basically a digitised Public Switched Telephone Network (PSTN) and it is replacing the analogue modem-to-modem connections in corporate WANs. Being a circuit-switched connection it can guarantee many QoS parameters such as jitter, bandwidth, and delay. For SOHO users Basic Rate Interface ISDN can offer the best price / performance connection. The only major problem related to ISDN is that it is a point-to-point connection. [Hall96]

Layer 1 based solutions are actually the same private networks that VPN is thought to replace. In addition to the high cost of using leased lines the lack of scalability is other major problem. A mesh topology is needed to interconnect multiple sites, because leased line connections are point-to-point. Administrating and expanding large-scale mesh topology can be a nightmare (Figure 3-2).

3.3.2 Link Layer (Overlay model)

Layer 2 VPNs can offer many benefits over leased line solutions. For example a Frame Relay connection is usually cheaper to use than a leased line connection. Two major technologies are used to build Intranet & Extranet VPNs: Asynchronous Transfer Mode (ATM) and Frame Relay (FR) (see Table 3-1). Layer 2 Tunnelling Protocol (L2TP) is used to create Access VPNs.

Layer 2 Tunnelling Protocol

L2TP is a standard tunnelling protocol that is implemented using the best features of Microsoft's Point-to-Point Tunnelling Protocol and Cisco's Layer 2 Forwarding (L2F) protocol. L2TP is used to tunnel Point-to-Point Protocol (PPP) frames over point-to-point (PP) links. PPP is used to encapsulate multiprotocol packets over layer 2 links (e.g. ISDN, ADSL, etc.). With this kind of a connection the layer 2 link and the PPP sessions end in the

same device, usually Network Access Server (NAS). L2TP enables the separation of layer 2 link and PPP sessions so that the end-points reside in different devices. Thus, L2TP enables a PPP session over a packet switched network, i.e. Access VPN over the Internet (Figure 3-4). [RFC2661]

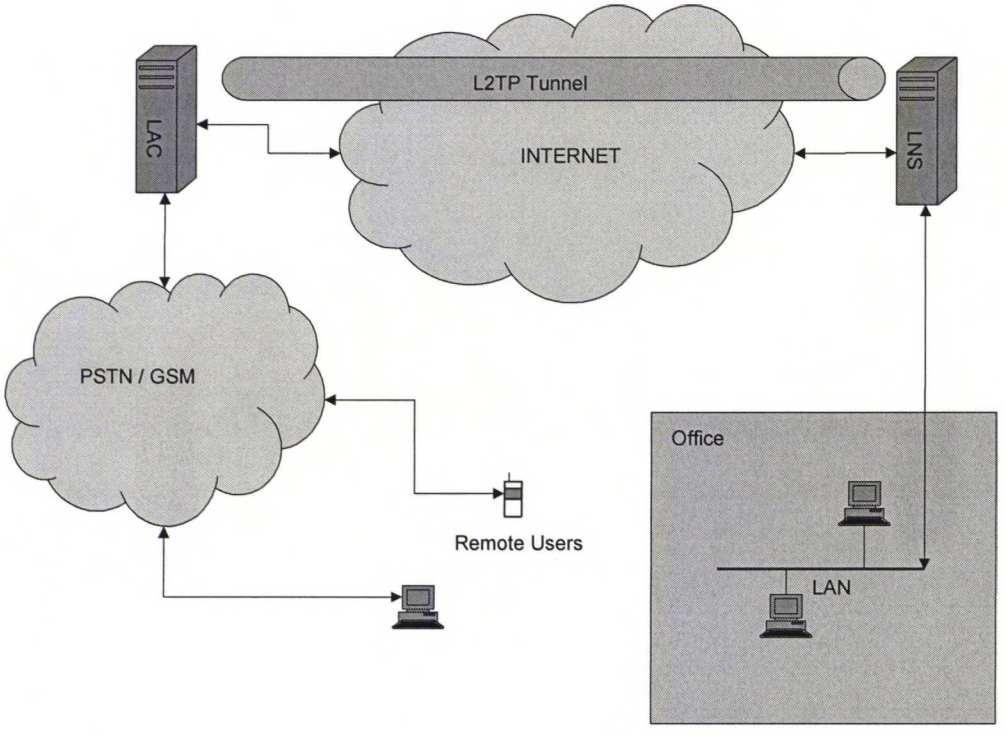


Figure 3-4: L2TP Tunnel

Remote user uses PPP to create a link to NAS (which acts here as L2TP Access Concentrator (LAC)). Then LAC creates a tunnel to Home Gateway (which acts as L2TP Network Server (LNS)) and the remote user has a connection to home network as if he/she is connected to the home network directly. User authentication is performed in home network by using Password Authentication Protection (PAP) or Challenge Handshake Authentication Protocol (CHAP). L2TP does not encrypt the data that it carries between LAC and LNS. But because L2TP is transferred over UDP/IP the security mechanism of IPsec can be used to protect the data on end-to-end connections. [RFC2661] L2TP can also be used to connect two LANs, even though it is mainly used to implement Access VPNs

[Kos98].

Frame Relay and Asynchronous Transfer Mode

ATM and Frame Relay are discussed together due to their similarities in VPN implementation. Both ATM and Frame Relay use virtual circuits in order to create VPNs. Also they are both widely deployed technologies. ATM and FR are International Telecommunications Union – Telecommunication Sector (ITU-T) and American National Standards Institute (ANSI) standards.

Frame Relay standard defines a process for sending data over public data network using packets (or frames). It is designed to be a pure and simple layer 2 protocol that relies on upper-layer protocols for error correction and flow control [FRForum]. Due to simplicity Frame Relay can offer efficient and high performance point-to-point connections, such as LAN-to-LAN. Frame Relay can be used to implement VPNs using Frame Relay Virtual Circuits (VCs). Frame Relay Virtual Circuit is a logical connection between two Data Terminal Equipment (DTE) devices over Frame Relay Packet-Switched Network (PSN). There are two types of virtual circuits: Switched and permanent. Switched virtual circuit (SVC) is a temporary connection between two DTEs. After the SVC is terminated it can be established again using signalling. Permanent virtual circuit (PVC), on the other hand, can be permanently configured between two DTEs using network management operations [Down98].

Asynchronous Transfer Mode is a high-speed connection-orientated technology designed to support wide variety of applications and services. Data is transmitted in fixed size cells (53 bytes) over a switch network. One of the advantages of ATM is that it can support differentiated QoS. ATM providers six service classes: Constant Bit Rate (CBR), Real-Time Variable Bit Rate (rt-VBR), Non-Real-Time Variable Bit Rate (nrt-VBR), Unspecified Bit Rate (UBR), Available Bit Rate (ABR) and Guaranteed Frame Rate (GFR). Thus, the service provider can offer end-users services with different pricing schemes by using these classes to differentiate VPN traffic. As in Frame Relay, virtual

circuits must be created across the ATM network prior to any data transfer. There are two types of virtual circuits: virtual paths, identified by virtual path identifiers (VPI); and virtual channels, identified by the combination of a VPI and a virtual channel identifier (VCI). There can be several virtual channels in one virtual path [ATMForum]

Virtual Circuits can be used to create logical paths over ATM/FR backbone (Figure 3-5). Logical paths are used to secure and separate the VPN traffic. Separation of different types of VPN traffic can be used to offer differentiated level of service to end-user, e.g. expensive golden VoIP and best-effort silver File Transfer Protocol (FTP) links. Virtual circuits enable basically the same level of security as leased lines. If service provider is not a trusted party then the customer must implement their own security architecture, e.g. IPsec.

ATM/FR based VPNs are quite common because both ATM and FR are so widely deployed and well-known technologies. VPNs can be created quickly and inexpensively using existing ATM/FR infrastructure. Both technologies can guarantee QoS on layer 2, thus they enable straightforward Service Level Agreement (SLA) model. However, IP-VPN providers usually require layer 3 SLAs.

The major problem with ATM/FR VPNs is very similar to leased lines. A full point-to-point mesh of virtual circuits is required if all branches must be interconnected and this can lead to scalability problems. Full VPN topology must be maintained in each Customer Premises Equipment (CPE). This means that maintenance and implementing changes in a large VPN can be complicated and time consuming.

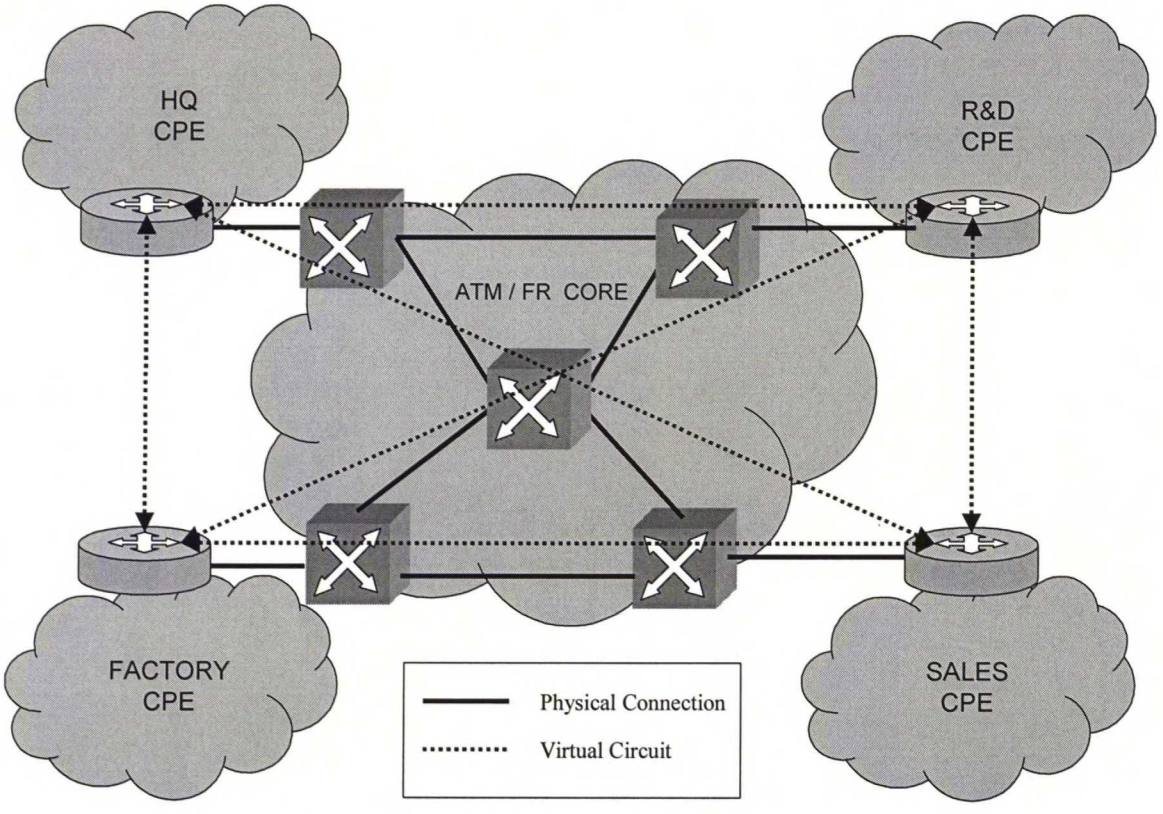


Figure 3-5: ATM / FR Network

3.3.3 Network Layer (Overlay model)

This thesis will concentrate only on VPN implementations that are based on the IP protocol. Other network layer protocols are out of scope of this Master's thesis.

Internet Protocol security (IPsec)

Internet Engineering Task Force (IETF) has developed a set of protocols that enable secure transfer of packets at the IP layer. This architecture is called Internet Protocol security (IPsec). IPsec is thought to be the protocol that enables network layer VPNs. IPsec is provided at the IP layer so it is transparent to the upper layers. [RFC 2401]

IPsec architecture provides confidentiality, integrity, and authenticity for an end-to-end connection at the network layer. IPsec uses two protocols to protect the data: IP Authentication Header (AH) and Encapsulating Security Payload (ESP). Both AH and ESP provide connectionless integrity, data origin authentication and optional anti-replay service whereas ESP may also provide confidentiality and limited traffic flow confidentiality. Therefore AH basically provides source authentication and integrity without encryption, while ESP provides authentication and integrity along with optional encryption. IPsec provides two modes of use: Transport mode and Tunnel mode. The transport mode only encrypts the payload, thus provides protection only to upper layers. The tunnel mode also encrypts the IP header so it is more secure than the transport mode. [RFC 2401]

Internet Key Exchange (IKE) has been chosen to be the standard method for negotiating Security Associations (SAs) and to exchange cryptographic keys between communicating entities. A security association is a simplex "connection" that affords security services to the traffic carried by it [RFC 2401]. Therefore bi-directional connection requires two security associations. IKE is a hybrid key management protocol that is used for authenticating the communicating peers and for providing authenticated keying material to security associations in a protected manner [RFC2409].

IPsec can be used to implement all the three VPN services: Intranet, Extranet and Access VPNs. Because IPsec does not have any QoS mechanisms it is not ideal in all VPN implementations. A good thing about IPsec, in addition to secure data transfer, is that the VPN service provider does not have to make any changes to their existing IP core network when using IPsec. However, IPsec based VPNs have the same scaling problems as layer 2 VPNs because IPsec tunnel is allegoric to a link layer virtual circuit.

Generic Routing Encapsulation (GRE)

Generic Routing Encapsulation is not the only encapsulation protocol that can be used to create VPNs. There are several other technologies such as [RFC1234], [RFC1226] and [RFC1853]. GRE is used as an example for encapsulation based VPNs since it is a generic

model.

GRE can be used to create tunnels over public data network [Figure 3-6]. It works in the same manner as IPsec tunnel but without security mechanisms. One advantage about GRE is that the VPN service provider does not have to make any changes to their existing IP core network. Encapsulated multiprotocol packets can be sent over existing IP network and only CPE equipment must be configured to support GRE. A mesh of virtual point-to-point interfaces must be configures so using GRE leads to same kind of scaling problems as with IPsec. GRE cannot guarantee QoS so application based QoS mechanisms are required. [RFC1701]

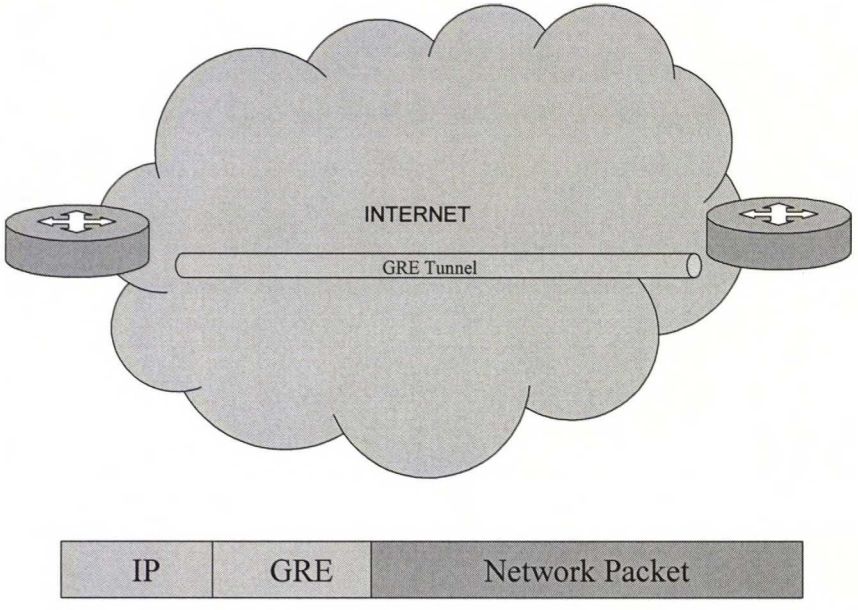


Figure 3-6: GRE Tunnel over the Internet

3.3.4 The Peer-to-peer Model

Access Control List (ACL) and Routing Tricks

It is also possible to create VPN by using facilities that are already present in routers and/or switches. These facilities include access control lists and other routing tricks. These techniques, however, do not provide any packet level security services. For example Cisco Content Service Switch (CSS) ACLs can be used to classify data traffic and permit only packets that have correct header information, like IP address range, port number and even application type. Access Lists (ALs) on the hand can be used to do basic traffic management. With access lists network administrator can enable Policy Based Routing (PBR) by using separate Edge Routers (ERs) for different customers. These techniques do not enable any network layer QoS services such as guaranteed delay and dedicated bandwidth but only offer best effort service. However, over provisioning is easier because PBR enables differentiation of IP packets, thus defining static routes to selected traffic. Even though these "tricks" enable VPN like connections they are not feasible solutions for large-scale VPN implementations. [Chap99]

Multiprotocol Label Switching (MPLS)

Introduction

MPLS is both a technology and an IETF specified framework for enabling label-based approach to packet forwarding. MPLS is used to tag IP packets with labels so that they are not forwarded but switched over a network. MPLS can be used to overcome many of the problems with today's IP networks, such as lack of QoS, security and traffic engineering, with minor chances to the existing network infrastructure. MPLS can be used with basically any link layer or network layer protocol, thus MPLS is called *multiprotocol.* [RFC3031]

There are many advantages for using label switching over traditional IP packet forwarding. Label switching is generally faster than IP packet forwarding so label switched packets travel quicker (and with shorter delay) over the network. Shorter delays in network elements result in less jitter. Small jitter is important in real-time applications such as Voice-over-IP (VoIP) and Video-on-Demand (VoD). MPLS also enables similar traffic engineering on network layer than e.g. ATM enables in link layer. On the other

hand, MPLS also overcomes some of the scalability and management problems related to virtual circuit based technologies (ATM, Frame Relay, etc). For example when a new site is added to an ATM network a new any-to-any virtual network must be provisioned and managed. MPLS based VPNs are based on networks – not connections. Hence MPLS is connectionless. On can say that MPLS combines the flexibility and scalability of IP and QoS and traffic engineering of ATM. [Blac01]

MPLS Key Concepts

Data traffic in MPLS network is transferred over Label Switched Paths (LSPs). LSP is basically a sequence of routers that defines a path from source (LSP Ingress) to destination (LSP Egress) (Figure 3-7). LSPs can be defined in one of two ways: prior to data transmission i.e. Ordered LSP Control or upon detection of packet i.e. Independent LSP Control [RFC3031]. When a packet arrives at the LSP Ingress, it is encapsulated into a MPLS Protocol Data Unit (PDU) and a MPLS label is attached to the MPLS header. Label is constructed based on the native data packet's header information (e.g. IP header), and at the LSP egress the MPLS header is removed and normal packet forwarding decision is made, e.g. normal IP forwarding based on IP address. Every node inside MPLS network reads and maps the label on the MPLS header and then switches it to the next node. Forwarding decision is based only on MPLS header, not on the network layer header information. MPLS label mapping is also called label swapping. There are three types of MPLS nodes: Ingress Label Switching Router (LSR), Transit LSR and Egress LSR. Ingress LRS and Egress LSR are also called Label Edge Routers (LERs) and transit LSR is called interior LSR. [Blac01]

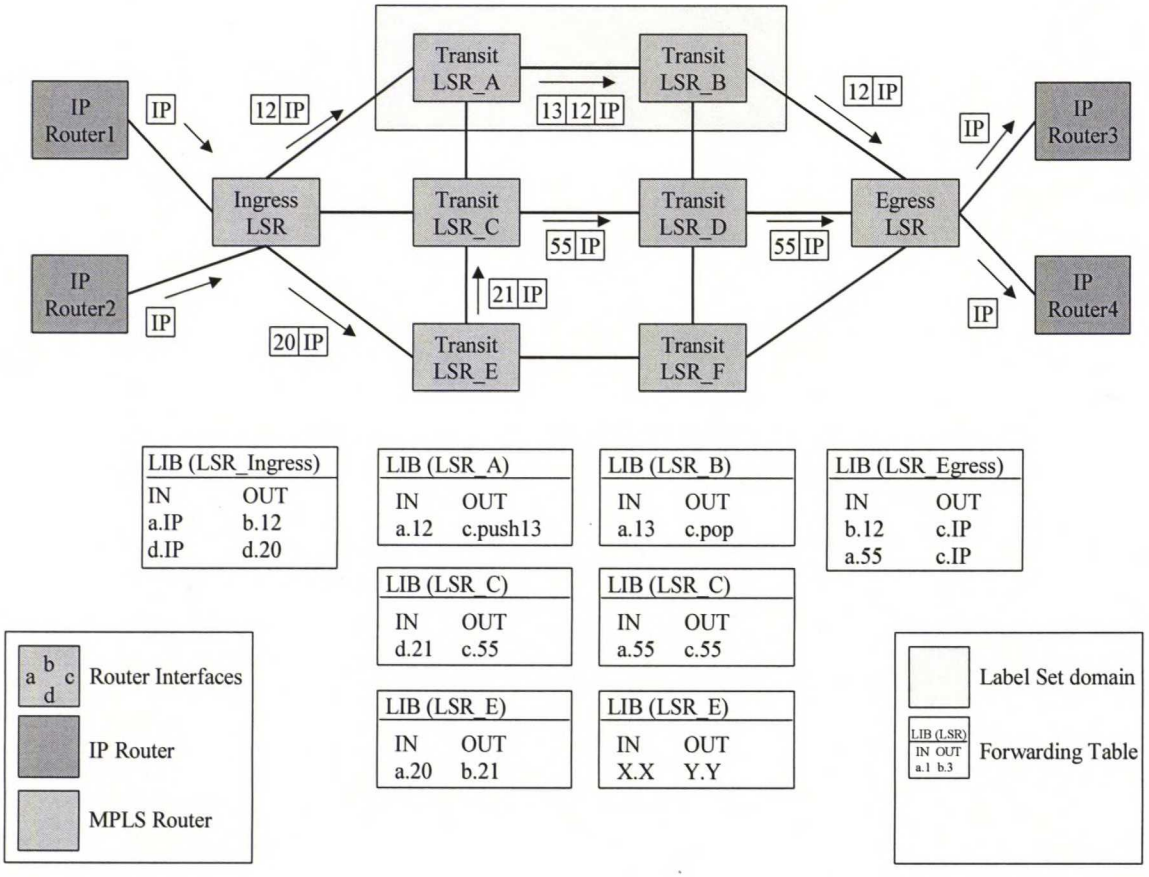


Figure 3-7: Label Swapping, Stacking and Forwarding

MPLS packets are grouped into classes called Forwarding Equivalence Class (FEC). FEC represents a group of packets that have the same transmission requirements over a MPLS network. All the packets belonging to a particular FEC are switched in the same manner en route to the egress LSR. The assignment of a packet to a particular FEC is done by the ingress LSR and it is done only once on the life cycle of a MPLS PDU. FEC enables differentiation of traffic streams thus it enables some QoS features to the data traffic. For example real-time voice traffic can have higher priority or different route than FTP traffic. FECs can be specified using information from other OSI layers. One probable solution is to use the Differentiated Services (DiffServ) information in IP header, i.e. Differentiated Services Code Point (DSCP), so that IP level QoS information can be used transparently in MPLS network. Every LSR has a Label Information Base (LIB) table that defines how the FEC-label bindings are executed. Thus it specifies how and where the packets must be forwarded. Every LIB entry contains one incoming label and one or more outgoing

labels. The label binding can be done locally or remotely. Local binding means that the LSR creates the binding by itself, whereas in remote binding the binding information is sent by other LSR using some label distribution protocol. [Davi00]

MPLS header is a sequence of fixed length labels. MPLS header is also called label stack. MPLS header is always located between link layer and network layer headers. A label is a locally significant identifier, which is used to identify the FEC that the packet is assigned to. Every label stack entry, i.e. label, is 32-bit long. There are four fields and MPLS label value occupies the first 20 bits. Next three bits are still experimental, but those could be used for example to map network layer QoS mechanisms, such as DSCP, to MPLS label. Stacking bit is used to indicate whether there is more than one entry in the label stack. This bit is set to one for the last entry in the label stack. The last 8 bits are reserved for Time-To-Live (TTL) field that defines the number of hops that the packet can still traverse in MPLS network. Figure 3-8 shows the generic MPLS label format in addition to some example uses depending on the underlying link layer protocol. [RFC3032]

The label binding decision between a label and a FEC can be done downstream or upstream. Upstream binding means that the LSR, which creates the label binding between the label and the FEC also sets the label to the packet that is sent. Thus, the LSR that created the binding is upstream with respect to the packet flow. In downstream binding the slow of information is opposite. MPLS architecture does not specify a single standard mechanism for label distribution. In fact, a number of different label distribution protocols are being standardized. When LSR creates or destroys a label binding the other LSRs can be informed with the new label binding information using several different methods. One method is to use existing protocols, such as Resource ReSerVation Protocol (RSVP), that are enhanced, i.e. piggybacked, to carry MPLS label information. IETF has also defined a new protocol called Label Distribution Protocol (LDP) especially for label distribution.

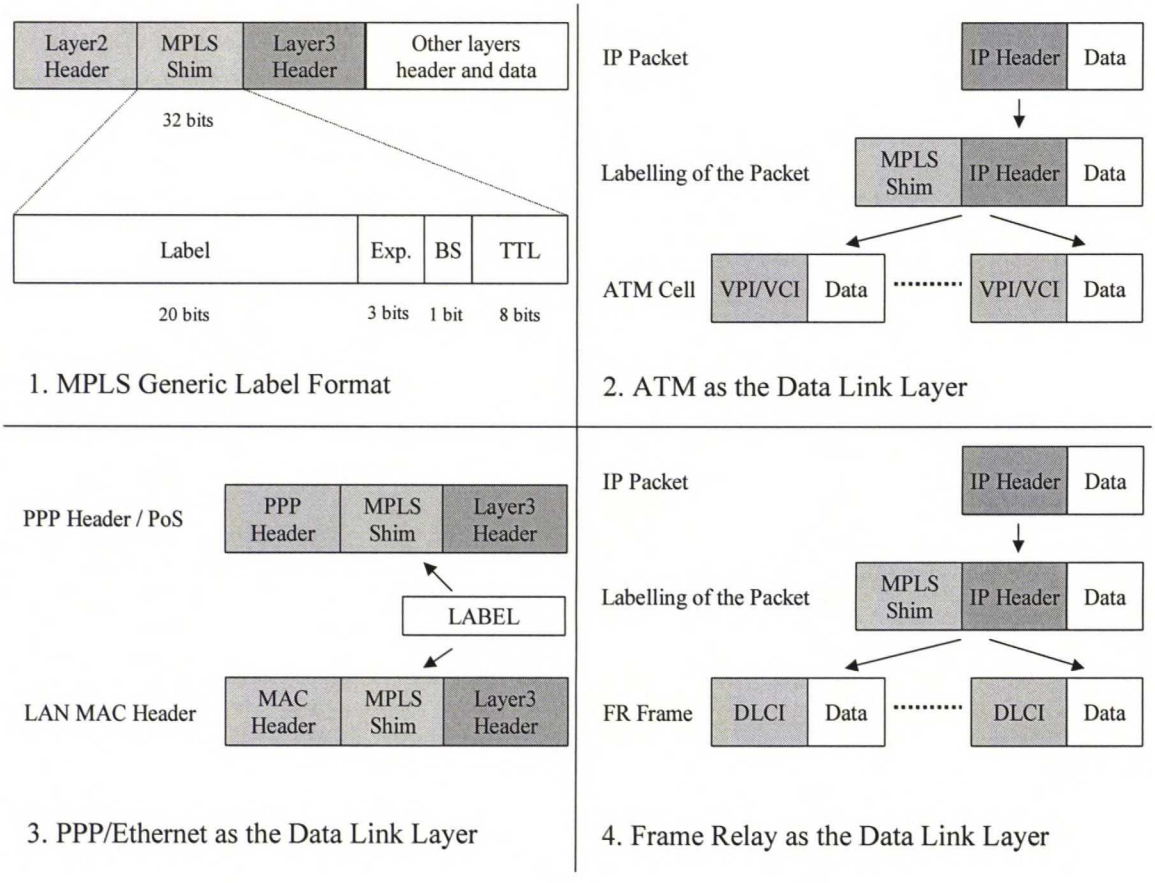


Figure 3-8: MPLS Header

LDP is a new protocol that is used to distribute the label binding information in MPLS network. LDP is a set of procedures and messages by which LSRs establish LSPs by mapping network layer routing information directly to data link layer switched paths. LDP associates a FEC with each LSP it creates. The FEC associated with an LSP specifies which packets are "mapped" to that LSP. There are four categories of messages that LDP currently supports:

- ξ Discovery messages: announce and maintain the presence of a LRS

- ξ Session messages: Establish, maintain and terminate sessions between LDP peers

- ξ Advertisement messages: To create, change and delete label mappings of FECs

- ξ Notification messages: To provide advisory information and signal error information

LDP is an evolving protocol and it is likely that new messages and procedures are added to the specification. [RFC3036]

Current routing protocols, such as RSVP and Border gateway Protocol (BGP) are enhanced to piggyback the label distribution information. Use of these protocols, however, is only possible in control-driven schemes. These protocols tie the label distribution to the distribution of routing information. This is not necessarily a drawback, because it ensures that the routing information is consistent with the label binding information, thus race conditions are avoided. This approach also simplifies the overall system operation by eliminating the need for a separate protocol for distribution of label information. [Davi00]

Traffic Engineering

One of the most interesting services that MPLS enables is traffic engineering on the network layer. The objective of traffic engineering is to enhance the overall network utilisation. Mapping the actual data to available resources, control the use of resources and enable rapid traffic re-routing in error situations are the key features that traffic engineering is able to provide. Current traffic engineering solutions in large IP networks are based on ATM. Because the router network is not aware of the ATM there are two control planes. MPLS enables a single IP level control plane that matched to the physical topology of the network. By using MPLS service providers are able to separate the IP level network traffic in an economic and scalable way so that they are able to provide different classes of services to customers. In MPLS the traffic engineering is provided by using explicitly routed paths. E.g. VoIP traffic can have a separate, high-speed path and FTP-traffic a low-speed path without QoS requirements. [RFC2702] Constrained-based routing (CR) protocols such as RSVP and CR-LDP can be used to enable dynamic traffic

engineering in MPLS network [Blac01].

Quality of Service

One of the most studied areas of IP networks is QoS, or actually the lack of it. Several protocols have been developed in order to enable all the QoS services that different types of applications require. There are basically two architectures that are used to enable QoS in IP networks. Integrated Services (IntServ) and DiffServ. RSVP is a signalling protocol that was developed to enable IntServ. It was developed to enable circuit-switched- like paths over packet-switched IP network. RSVP fulfils the design requirements and has been used successfully in many networks. However, there are some problems related to RSVP. RSVP basically works by creating micro flows over a network. Each router on the RSVP route must maintain information about the micro flow. If there are several RSVP connections the management information may grow too large and the router is not capable of operating efficiently. For example in a backbone router there can be millions of micro flows and it is clearly impossible to maintain the status information of all these connections. The other problem with RSVP is that it uses soft state connections, thus the status of the connection must be updated regularly. This means that the amount of management data traffic can grow too large. DiffServ on the other hand uses DSCP to mark IP packets so that they are treated differently in a router. E.g. an IP packet with higher-class gets higher priority on the router queues, thus it traverses quicker through the network. The problem with DiffServ is that it cannot guarantee any QoS parameters, thus it is not adequate e.g. for real-time applications.

MPLS is though to be the magic bullet that finally enables end-to-end QoS connections. This however is a common misconception. The QoS model that MPLS delivers should be more like an enabler of IP based QoS than a new type MPLS based QoS. Most of the emphasis has actually been put on enabling the IntServ and DiffServ on MPLS connections. Thus we have to look at the MPLS QoS from the RSVP's and DiffServ's point of view. MPLS can ease the management burden of RSVP connections by creating the RSVP micro flows a bit coarser. E.g. the smallest size of a micro flow is the size of a

LSP. This means that the number of micro flows is less because several IP level RSVP connections can be bundled in to one LSP. The other thing is that MPLS has features that can be used to decrease the amount of management traffic. This is enabled by a reliable transfer mechanism and longer refresh interval for the management data. On the DiffServ model the experimental bits of MPLS header can be used to map the DiffServ class so that data traffic can be separated to different classes. [Davi01]

Even though MPLS is not able to provide guaranteed QoS services, it has some features that clearly are beneficial. The most important feature is that MPLS can be used to enforce IP QoS on the link layer. MPLS also supports the DiffServ classes and helps to address the scaling problems of IntServ model. The essential thing in the area of QoS is that MPLS supports the QoS models that service providers have implemented on the network layer.

BGP/MPLS VPN

There are several ways to use MPLS in order to build VPNs. One commonly used way is called BGP/MPLS VPN. As the name implies BGP is used for distributing routing information and MPLS is used for forwarding packets over the service provider's core network. An example network is shown in Figure 3-9. The network consists of following equipment: Customer Edge (CE), Provider Edge (PE) and Provider (P) devices. In general CE device can be expected to be a router, although it can also be a switch or even a single host. The customer may administrate their own CE devices so that service provider does not have access to them. CE device is called *managed CE* if the service provider is responsible of managing the device. If CE device is a router then it is a routing peer to a PE device. PE device is a router and it is connected to one or more VPNs via CE devices that are part of some VPN. PE and CE devices exchange routing information using some Interior Gateway Protocol (IGP), such as Open Shortest Path First (OSPF). Every PE device has a separate forwarding table for each VPN it is connected to. For example in figure 3-9 PE_1 has two forwarding tables. Every forwarding table has information only about those routes that are part of the same VPN. This enables overlapping address spaces between separate VPNs. It also restricts the communication between different VPNs. Thus

it enables link layer level security. In addition to PE devices there are Provider (P) routers in the service provider's core network. P routers are not attached to CE devices and they do not maintain any routing information of any VPNs. Only those PE devices that are connected to a particular VPN maintain the routing information about the VPN. Because CE devices are not routing peers to other CE routers at other sites of a particular VPN the routers at different sites do not directly exchange routing information with each other. Thus, CE devices can support VPNs with huge numbers of sites because the CE device is a routing peer only to that PE device that it is attached to. [RFC2547]

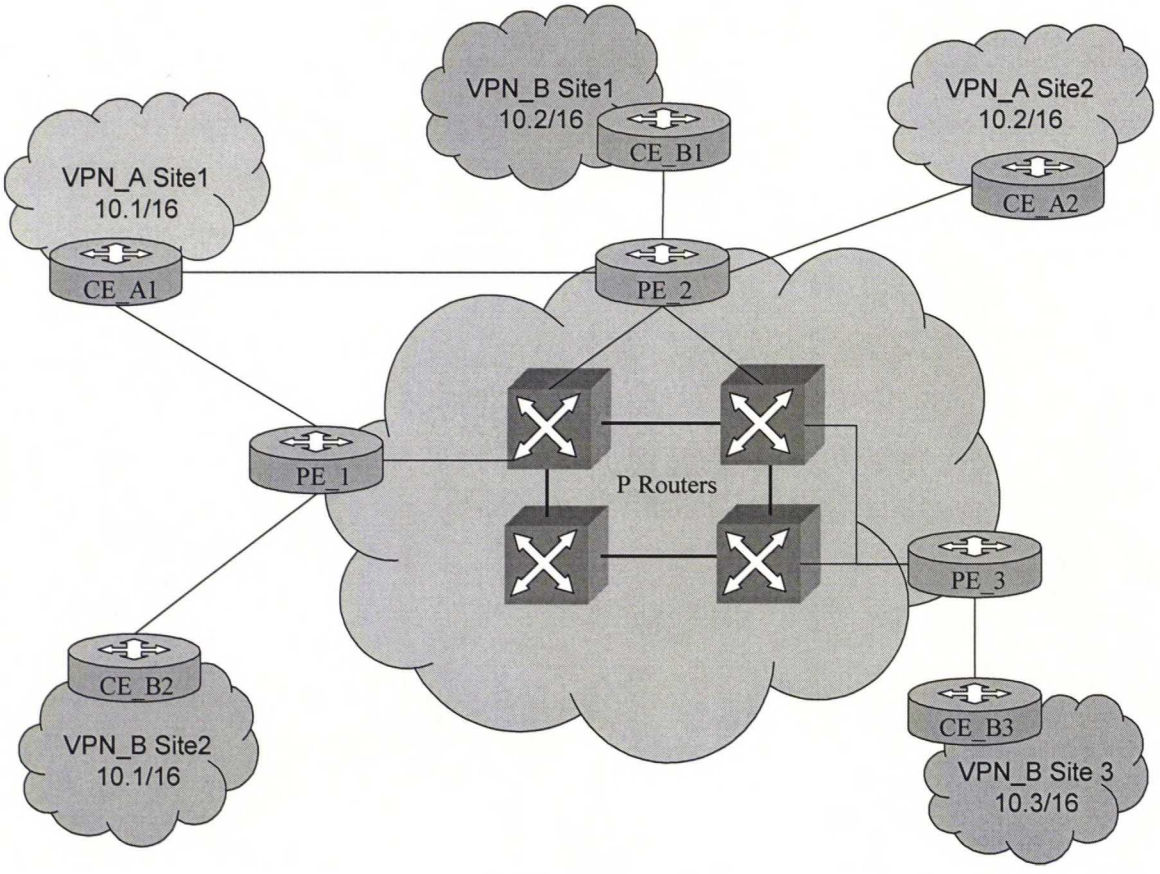


Figure 3-9: BGP/MPLS VPN

VPN sites are considered to be external entities, so BGP is chosen as the Exterior Gateway Protocol (EGP) that is used to create links between VPN sites. PE routers distribute VPN routes to each other using BGP. A BGP speaker can only install and distribute one route to a given address prefix. But because each VPN can have its own address space, which

can overlap with other address spaces used by other VPNs, BGP must be able to install and distribute multiple routes to a single IP address prefix. One must also make sure that the right VPN route is in the right forwarding table in order to eliminate the possibility that packets from one VPN are not forwarded to another VPN. These requirements are met by introducing a new address family, called VPN-IPv4 address family. It is defined using BGP Multiprotocol Extensions [RFC2283]. VPN-IPv4 address has an 8-byte prefix called Route Distinguisher (RD) and a 4-byte suffix that is the actual IPv4 address. RD is used to separate similar IPv4 addresses from one other. It can also be used to create multiple routes to a single VPN site. E.g. it might be wise to have a different route to a site from an intranet than from an extranet for security reasons. E.g. all extranet traffic is sent through a firewall. [RFC2547]

MPLS is used as the forwarding mechanism. BGP/MPLS VPN uses a two-level label stack to forward packet over the core network. This enables the scheme where P devices are not aware of the VPN routes. PE device acts as an edge LSR and when it receives a packet from a CE device, it selects the correct forwarding table to look-up the packet's destination address. If the destination CE device is attached to this PE device then it sets the packet directly to that CE device. If the destination CE device is attached to another PE device, then the VPN route label i.e. BGP next hop is pushed onto the packet's label stack as the bottom label. Then PE device looks the actual LSP trough the service provider's core network and sets the appropriate label as the top label. LSPs across the service provider's core network are established using some label distribution protocol, such as LDP or RSVP. LDP is used to establish a best-effort LSP between two PE routers. A constrained-based routing protocol or traffic engineering can be used to enable LSPs with QoS features. The MPLS then carries the packet to the destination PE device, which pops the top label and uses the bottom label to forward the packet to the correct CE device. [RFC2547]

One of the most important feature of MPLS based VPN solution is that it is highly scalable. If we consider the scalability of a BGP/MPLS VPN solution we can find a few important points. First point is that P devices are not aware of the VPN routes. Actually P devices do not even have to be aware of the CE devices. Also a CE device, within a given

VPN, maintains routing information only about its routing peer (i.e. PE device) that it is directly connected to. It does not have any routing information about the other CE devices within the VPN. Thus the routing information in a CE device is constant and independent of the number of VPN sites in a VPN. In the overlay model the CE device must maintain routing information of all of the VPN sites that it is connected to. This means that in a case of fully meshed VPN the routing information grows proportionally as the number VPN sites grows. Also if new VPN sites are added or deleted in the VPN the routing information must be updated to all of the interconnected CE devices. In peer-to-peer model only the PE device that is connected to the VPN site requires configuration. Thus, the scalability of the peer-to-peer model in the area of configuration management is superior to overlay model. In addition the PE devices require fewer configurations because they maintain information only about the VPN routes that they serve, thus they must maintain routing information only about the PE devices that are connected to the same VPN. However, some times the number of VPN sites can grow so high that PE devices are not able to support all the VPN routes between the PE devices. In this kind of a situation a BGP Route Reflector (RR) can be used to eliminate the need to maintain a full mesh of BGP session between PE devices within a VPN. If some PE device has too many CE devices to maintain, then an additional PE device can be added and the workload can be balanced. All this means that there is no single network element that has to maintain routing information about all the VPN routes that the service provider provides. Thus, the number of VPNs and VPN sites is not limited by the capacity of any individual component. [RFC2547]

3.4 Conclusions

VPNs based on the overlay model are predominant today. However there are lots of problem related to these kinds of VPNs when talking about large-scale VPN solutions. First of all the VPN customer must be able to design, build, configure and maintain the VPN backbones of such solutions by themselves. The customer requires a huge amount of knowledge not only about IP routing and IP QoS but also about link layer and about the co-operation of link layer and network layer. Service provider can offer so-called managed router- service in order to ease the management burden of the customer. In this kind of a

solution the service provider is responsible for the design and maintenance of the VPN backbone by offering each customer virtual backbones. However, this leads to a situation where service providers face the problems. A service provider must be able to design and maintain huge amount of virtual backbones. Let us say that the service provider has 10,000 VPN customers. Then the service provider must be able to administrate at least 10,000 virtual backbones. If the customers have several sites that require mesh topology then the number of virtual backbones can grow exponentially. The amount of configuration information leads to a situation where even small changes to existing VPN can require huge amount of work. Basically this means that service providers are not able to support large numbers of customers at reasonable costs. In addition to the problems mentioned above there are additional problems related to VPNs based on IPsec and GRE. These network layer technologies are not able to guarantee the level of QoS that LL and FR/ATM users are used to. [Davi00]

The VPNs based on peer-to-peer model are able to overcome some of the fundamental limitations that are related to overlay model. The main problem with the overlay model is the scaling problem. Peer-to-peer model enables service provider to have huge numbers of VPN customers with many interconnected sites. VPN solutions based on MPLS enables secure and highly scalable VPNs with end-to-end QoS when using RSVP. MPLS labels isolate the IP addresses within public network from customer IP addresses. This means that the customer equipment is isolated from the public IP network, thus connecting to a MPLS based VPN solution do not require any changes in the customer's internal equipment except for the CE device. On the other hand the service provider is able to sell VPN solutions that supports IP QoS on the link layer. Also the Service Level Agreement (SLA) offerings can be implemented on IP level. Service provider needs only one network infrastructure because no dedicated equipment is assigned to customers. [Davi00]

As we have seen, there are several ways to implement VPN service. However, when talking about economical, large-scale and maintainable VPN solution it seems that MPLS is the best solution from the service provider's point of view. It has basically all the advantages that but no clear drawbacks. There seems to be only advantages in MPLS based

VPN. It is difficult to predict the long-term future, but it seems that MPLS could be the key technology that enables service-driven IP networks. Table 3-2 summarises the key features related to VPN

Table 3-2: VPN solutions

	Any-to-Any Complexity		QoS	Privacy	Cost of usage
	User	Network			
Leased Line	N^2	N^2	Yes	Yes	High
FR / ATM	N^2 (logical)	N^2 (logical)	Yes	Yes	Medium
IP (IPsec, GRE)	N	N	No	No	Low
MPLS	N	N	Yes	Yes	Medium

4 Provisioning of MPLS based VPN

4.1 Introduction

MPLS based VPN solution was chosen as an example VPN solution for studying the provisioning requirements. There are two main reasons for this selection. First reason is that the MPLS based VPN is one of the most asked technologies that service providers require mediation system vendors to support. The other reason is that MPLS based VPN is maintained usually by service providers that rely on automated service activation tools, such as mediation devices. MPLS based VPN can also be used to identify the most important interfaces, such as LDAP and router interface.

This chapter describes an example MPLS based VPN solution. This network is used to identify the provisioning requirements that MDS/SAS must be able to fulfil. The network is composed of service provider's core MPLS network and VPN customers' CE devices. This VPN example is based on a BGP/MPLS VPN solution and it is a general example how MPLS network can be implemented.

4.2 Topology

4.2.1 Physical Topology of the Core Network

Physical topology of the service provider's core network is shown in figure 4-1. The MPLS core network is built using eight ATM switches (P devices) that are interconnected via OC-3 (155 Mbps) links. There are also eight PE devices at the edge of the core network. These routers are connected to the CE devices via Ethernet interfaces. The PE-to-CE interfaces are pure IP connections. The core network is distributed over a number of cities.

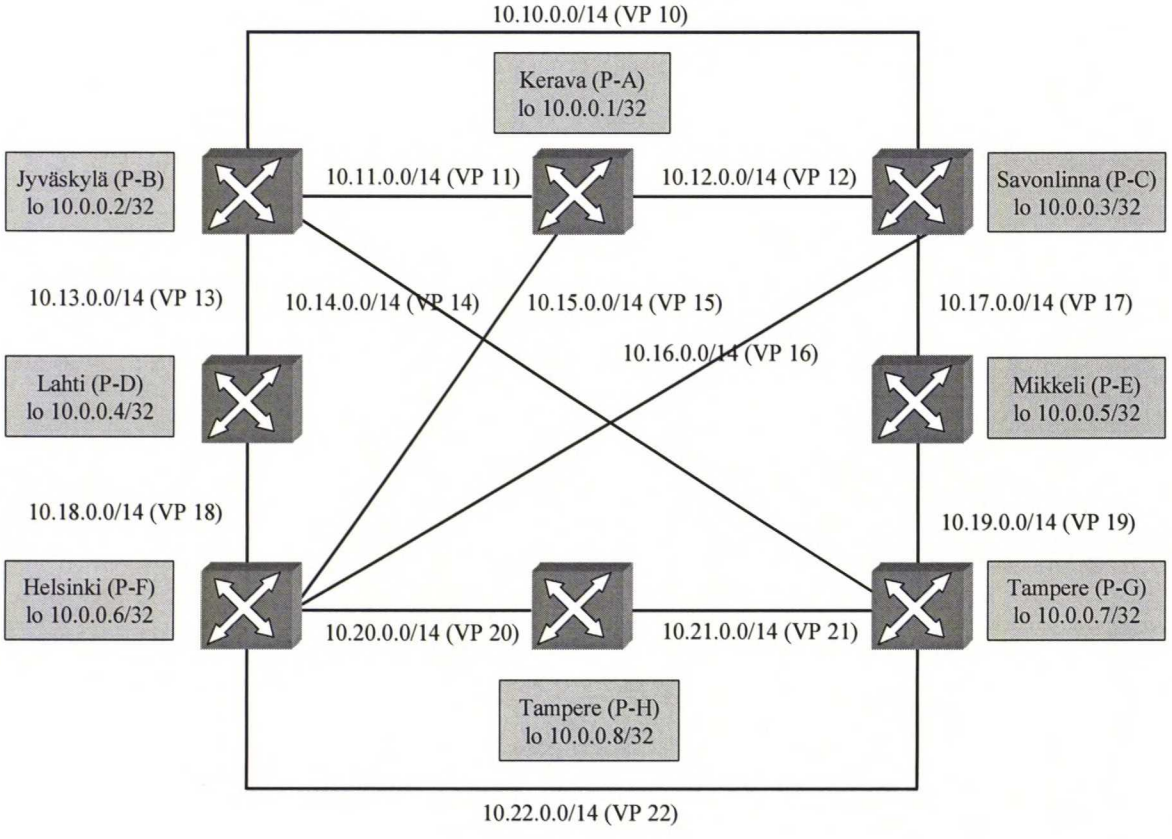


Figure 4-1: Physical Topology of the MPLS Core Network

4.2.2 VPN Topology

There are three VPNs in the service provider's network. MDS/SAS will be used to activate the VPN_A between the sites in Kerava_1, Kerava_2, Helsinki and Tampere. VPNs are configured so that they can have their own address spaces. Additionally the data traffic is restricted inside VPNs so inter-VPN connections are not possible. Below is a description of the network management information that is required to configure the VPNs.

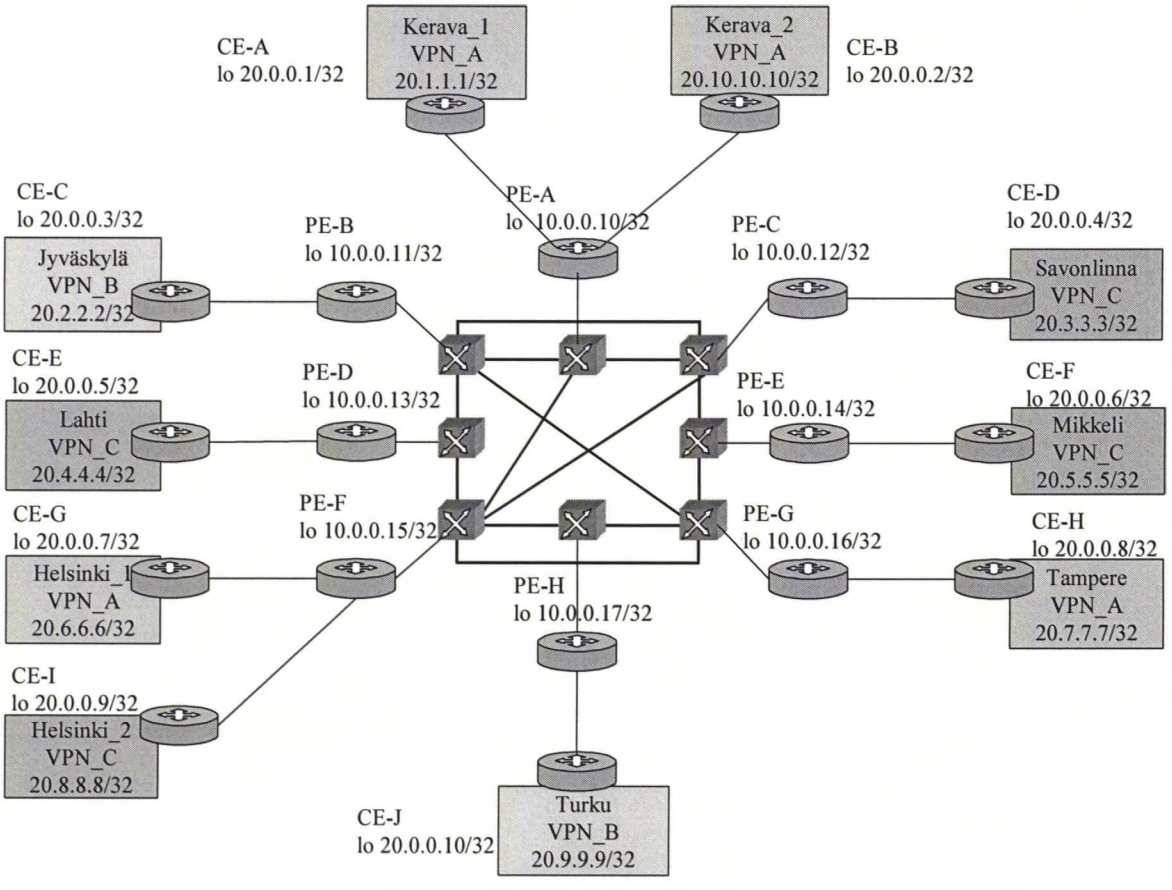


Figure 4-2: VPNs in the Service Provider's Network

Table 4-1: IP Addresses

PE devices	(NE / IP Address)		
PE-A	P-A / 10.30.0.0/14	CE-A / 20.1.0.0/14	CE-B / 20.2.0.0/14
PE-B	P-B / 10.31.0.0/14	CE-C / 20.3.0.0/14	
PE-C	P-C / 10.32.0.0/14	CE-D / 20.4.0.0/14	
PE-D	P-D / 10.33.0.0/14	CE-E / 20.5.0.0/14	
PE-E	P-E / 10.34.0.0/14	CE-F / 20.6.0.0/14	
PE-F	P-F / 10.35.0.0/14	CE-G / 20.7.0.0/14	CE-I / 20.8.0.0/14

PE-G	P-G / 10.36.0.0/14	CE-H / 20.9.0.0/14	
PE-H	P-H / 10.37.0.0/14	CE-J / 20.10.0.0/14	

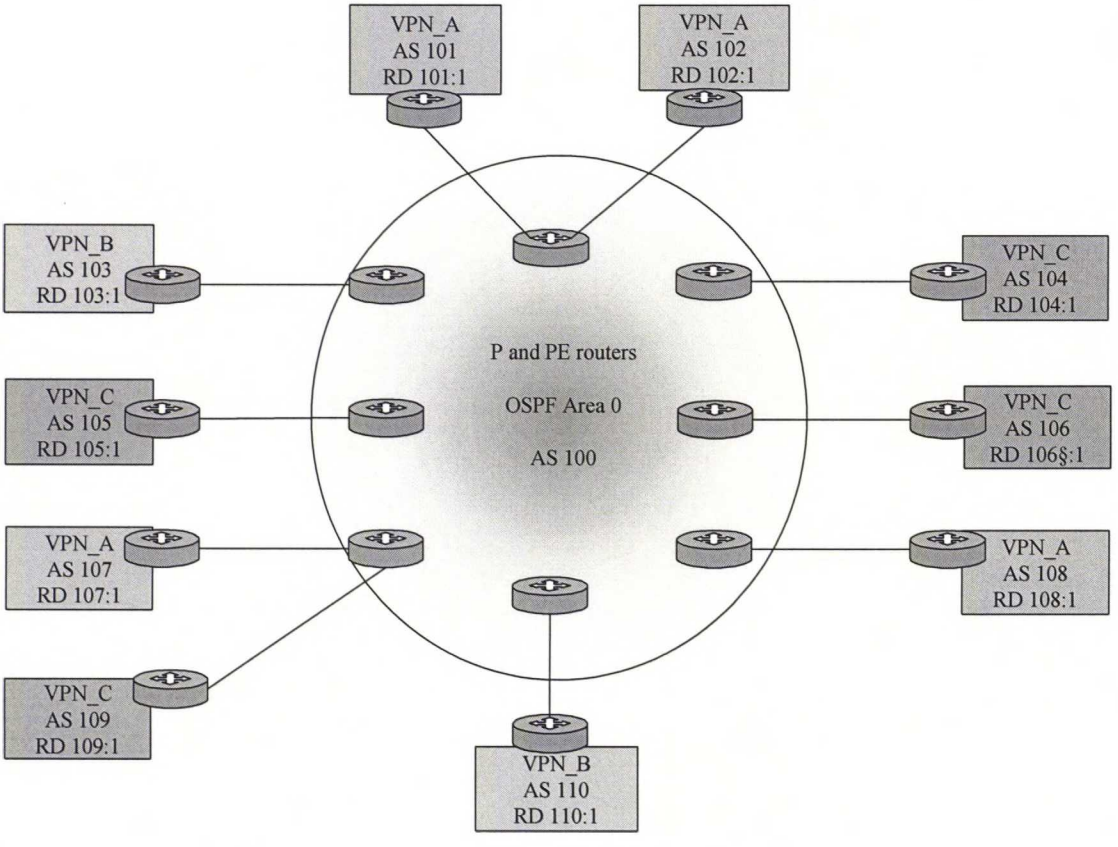


Figure 4-3: OSPF Area 0 and Autonomous Systems

The core network is configured to propagate the routing information using OSPF. When multiple areas are involved in one network the OSPF protocol requires that one of these areas must be area 0. It is called backbone and it has to be at the centre of all other areas, i.e. all areas are physically connected to the backbone. Figure 4-3 shows the Autonomous Systems (ASs) that are assigned to the BGP routers (CE and PE devices). The PE devices are configured to share information about neighbouring networks that are part of the same VPN. This solution do not use BGP Route Reflectors so the PE devices are required to

maintain routing information of all the other PE devices in particular VPN. [RProt]

4.2.3 Management Topology

Cisco VPN Solution Center (VPNSC) manages the core MPLS network and some of the CE devices. VPNSC cannot be used with all of the CE devices because the VPN customers manage some of the CE devices by themselves and one site has an Enterasys router. VPNSC is only capable of supporting Cisco products. MDS/SAS is used to manage the CE devices by sending email to the customer's network engineers or by configuring the router.

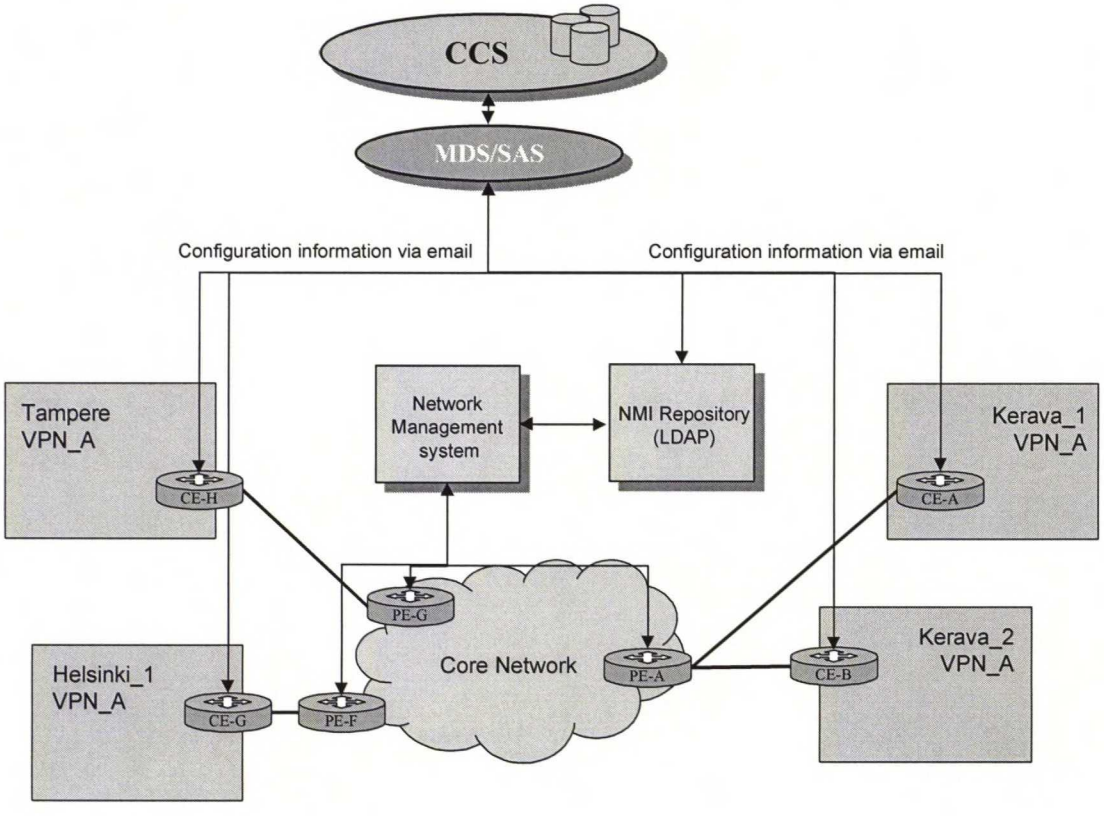


Figure 4-4: Management Topology

Table 4-2: Network Elements

Equipment	Type	Software	Vendor
P devices	Catalyst 8510 MSR	IOS 12.0(4a)W5(11a)	Cisco
	Catalyst 8540 MSR	IOS 12.0(13)W5(19)	Cisco
PE device	7200	IOS 12.0.7 (T)	Cisco
	7200	IOS 12.1 T	Cisco
CE device	7200	IOS 12.0.7 (T)	Cisco
	7120	IOS 12.0.7 (T)	Cisco
	7140	IOS 12.1 T	Cisco
	3660	IOS 12.0(5)T	Cisco
	X-Pedition SSR-2-B128	SSR firmware 1.2.0.0	Enterasys

4.3 Configuration

Basic MPLS Configuration

The core MPLS network is first configured to support MPLS. The configuration can be done manually or by using Network Management System (NMS). In our case the core network is configured using VPNSC. Only the service provider's core network needs to be configured for MPLS because the interfaces between PE and CE devices are pure IP. The basic configuration requires following tasks in addition to normal router configuration [MPLS01]:

1. First Cisco Express Forwarding (CEF) is enabled in all the routers and switches that are in the service provider's core network.

2. OSPF is used to distribute the routing information in the core network. All the network elements in the core network run OSPF in area 0.

3. Routers in the service provider's network are configured incrementally throughout a network by enabling MPLS in all of the interfaces between P and PE routers.

4. BGP sessions are set-up between all the PE routers that are part of the same VPN. OSPF routing information is distributed into BGP on the PE devices and BGP routing information to OSPF.

MPLS is deployed throughout the core network after these steps, thus all core routers are capable of forwarding MPLS packets. Also the protocols for distributing the routing information are activated. MDS/SAS can be used to configure the core network. However, it is more likely that MDS/SAS will be used only when the actual VPN routes are activated. Following chapter describes how a VPN is activated.

MPLS VPN Configuration

Only the PE and CE devices require configuration whilst activating a VPN. P routers in the core network are not aware of the VPN routes nor of the CE devices so there is no need to configure them. The activation is straightforward on the customer site. The only requirement is to enable a BGP session on the CE device. On the service provider's network the PE devices require a bit more configuration. I will use CE_G and PE_F routers as examples how the configuration is done (figure 4-2). [MPLS01]

1. Two VPN routing/forwarding instances (VRFs) are defined for the PE_A router because it is connected to two VPNs. Both VPN_A and VPN_C are associated with one of the VRF. The VRF defines the VPN membership of a customer site attached to a PE router. Each VPN is associated with one or more VRFs, however each VPN site can be associated with only one VRF.

2. Then router interfaces are set-up as VRF links to CE routers. In our case

Ethernet interface is set-up as a VRF link to a CE router.

3. The BGP sessions for the PE-to-CE device pairs are configured. BGP is used to distribute the routing information between the CE and PE devices. Also other options, such as static routing or OSPF, could be used to propagate the routing information.

Now the VPN_A is activated on CE_G and PE_F. All the other PE and CE devices that are part of the VPN_A must be configured similarly in order to activate the whole VPN. Note that adding a new VPN site does not affect the configuration of the other CE devices. Only the PE device configuration must be updated so that they are aware of all of the VPN routes. The configuration information is shown in appendix 3.

4.4 Network Management Information Repository

Customer wants to maintain VPN customer information in a LDAP directory. The LDAP directory schema has been extended to support required customer information. The directory contains basic network management and customer accounting information. The Directory Information Tree (DIT) is based on an IETF draft, thus the directory name is of format dc=company, dc=com. Figure 4-5 depicts the DIT and the object classes.

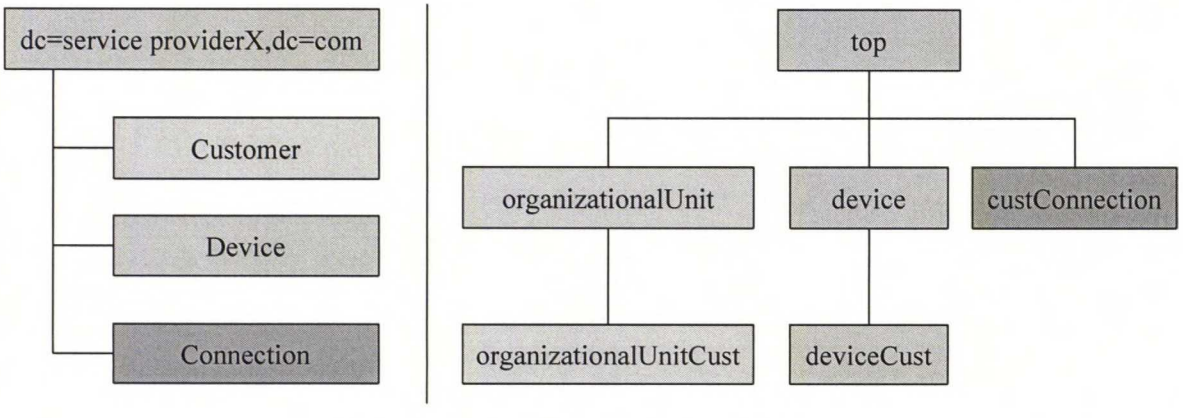


Figure 4-5: DIT and Object Classes

The representation of the DIT above provides an overview of the tree structure. The object classes and their attributes within this structure are defined below. This description does not take any stand on the security of the directory. MDS/SAS is considered to be a client with full operator level access to the directory. Only those attributes that are used are described.

Customer branch is used to maintain customer related administrational information, such as contact person information and activation dates of the VPN links. Device branch can be used to save CE device information in situations where CE device is managed device. Connection branch is used to save information about the CE-to-PE connection. Service provider uses the same LDAP directory to maintain other information also. E.g. service provider's internal organisational information and network management information. MDS/SAS is only responsible for maintaining the information mentioned in this chapter.

Table 4-3: Customer Branch (organizationalUnit)

DN: vpnName=VPN X, ou=customer, dc=service provider X, dc=com		
Attribute Name	**Description**	**Type**
Required Attributes:		
objectClass	Defines the object classes for the entry.	
ou	The name of the customer.	cis
Allowed Attributes:		
description	Text description of the customer.	cis
fax	Fax number of the customer.	tel
internationalIsdnNumber	Customer ISDN number.	tel
l	Physical location of the customer.	cis
postalAddress	Customer mailing address.	cis
postalCode	Customer postal code.	cis
postOfficeBox	Customer post office box.	cis
seeAlso	URL to information relevant to the customer.	dn
st	State where the customer resides.	cis
street	Customer street address.	cis
telephoneNumber	Customer telephone number.	tel
userPassword	Password with which the entry can bind to the directory.	bin

Table 4-4: Customer Branch (organizationalUnitCust)

DN: vpnName=VPN X, ou=customer, dc=service provider X, dc=com		
Attribute Name	**Description**	**Type**
Required Attributes:		
objectClass	Defines the object classes for the entry.	
vpnName	The name of the VPN.	cis
vpnSiteName	The name of the VPN site	cis
Allowed Attributes:		
vpnStatus	Status of the VPN service (e.g. ready).	cis
activationDate	Activation date of the VPN.	cis
deactivationDate	Deactivation date of the VPN.	cis
slaInformation	URL of the SLA information.	ces
contactPerson	Name of the customer contact person	cis

Table 4-5: Device Branch (device)

DN: vpnName=VPN X, cn=device name, dc=service provider X ,dc=com		
Attribute Name	**Description**	**Type**
Required Attributes:		
objectClass	Defines the object classes for the entry.	
cn	The name of the CE device.	cis
Allowed Attributes:		
l	Physical location of the CE device	cis
ou	Customer to whom the CE device belongs to.	cis
owner	Name of the CE device owner.	cis
seeAlso	URL to information relevant to the customer	dn
serialNumber	Serial number of the device	ces

Table 4-6: Device Branch (deviceCust)

DN: vpnName=VPN X, ou=device, dc=service provider X ,dc=com		
Attribute Name	**Description**	**Type**
Required Attributes:		
objectClass	Defines the object classes for the entry.	
deviceId	Unambiguous id of the CE device.	ces
peerId	Unambiguous id of the connected PE device.	ces
vpnName	The name of the VPN.	cis
vpnSiteName	The name of the VPN site	cis
ipAddress	IP address of the CE device	cis
Allowed Attributes:		
deviceVendor	Name of the equipment manufacturer.	cis
deviceModel	Model of the CE device.	cis
deviceOS	Os version of the CE device.	cis
StatusInfo	CE device status information.	cis
macAddress	MAC address of the CE device.	cis

Table 4-7: Connection Branch (connectionCust)

DN: vpnName=VPN X, SubnetNumber =connection, dc=service provider X ,dc=com		
Attribute Name	**Description**	**Type**
Required Attributes:		
objectClass	Defines the object classes for the entry.	
SubnetNumber	Subnet address of the CE-to-PE connection.	cis
SubnetMask	Subnet mask of the CE-to-PE connection.	cis
vpnName	The name of the VPN.	cis
vpnSiteName	The name of the VPN site	cis
Allowed Attributes:		
addressType	Address type CE-to-PE connection.	cis
protocolType	Protocol type of the CE-to-PE connection.	cis
lanType	LAN type of the CE-to-PE connection.	cis
routingProtocol	Routing protocol used between CE and PE devices.	cis

4.5 Provisioning Interfaces

4.5.1 Router Interface

There are several ways to configure a router. One way is to use Simple Network Management Protocol (SNMP) to configure the Management Information Base (MIB) parameters. The problem with this approach is that some of the MIB parameters are read-only and cannot be modified using SNMP. Thus it is impossible to propagate all required information to the router.

There are also new protocols being developed in order to enable secure and reliable router configuration. One of these protocols is Common Open Policy Service (COPS). The problem with COPS is that it is not that widely used and routers are not yet able to act as

COPS clients. The actual solution requires an agent acting as a COPS client, which would use Command-Line Interface (CLI) in order to configure the router.

The only feasible solution is to use CLI. CLI is not an ideal interface to a network element and it definitely is not designed to be used by mediation devices, but by human administrators. However, CLI is an easily implemented interface and MDS/SAS can be configured to use Expect Macro Server in order to perform router configuration. Because Cisco is so dominant player in the router marker the other router vendors have implemented their routers so that their CLI implementations more or less emulate Cisco IOS CLI.

4.5.2 Customer Interface

Customer must be notified about the router configurations if the service provider does not have rights to configure the customer's CE devices. MDS/SAS can be configured to send the configuration information to the customer via email. The customer is then responsible for configuring the CE device by itself. If the CE device is a managed CE device (i.e. the operator maintains the device) then MDS/SAS can be used to configure the router. MDS/SAS is a good solution e.g. when the NMS does not support the type of CE device the customer is using. E.g. in our case the VPNSC is only capable of supporting Cisco routers.

4.5.3 Network Management Information Repository Interface

LDAP stand-alone server is used to save the network management information. The LDAP directory has several object classes to save the network management information and some of these classes must be maintained by MDS/SAS. The VPN customer information is stored in object classes described earlier.

MDS/SAS uses the network element interface module to interface with the LDAP

directory. The implementation of this interface is part of this thesis. More about this interface can be found on chapter 6.

4.5.4 Conclusions

MDS/SAS can be used to interface with basically any network element in any type of network. However, because MDS/SAS is designed to operate in a telecommunication network the functionality of MDS/SAS is not optimal for many of the IP services. This is due to the differences between the management architectures of telecommunication and IP networks. Telecommunication networks use circuit-switched connections and these are established using some signalling protocol. Mediation device is not required to maintain, establish, or monitor these connections. The only function of mediation device is the transfer the required subscriber information to selected network elements.

IP networks use packed switched approach to data transfer. Activation of a connection may require configuration of several network elements. Usually service providers expect MDS/SAS to be able to do the required configuration. For example if service provider wants to offer BGP/MPLS VPN service to customers then MDS/SAS must be able to configure packet forwarding in all of the network elements in the service provider's core network. In general the problem is not with interfacing or configuration of these network elements but with the fact that MDS/SAS is not designed to work in these kinds of situations. Service providers expect MDS/SAS to be able to maintain this information and act as a network management system. However, MDS/SAS does not maintain network level information, such as subscriber or router status information. MDS/SAS only maintains location and connection information about the network elements itself. MDS/SAS is a great tool for interfacing different types of network elements. The problem is that MDS/SAS does not have the needed facilities to act as a network management system.

In order for MDS/SAS to act as a NMS, one should make following modifications. First of all the GUI should be changed so that it is more flexible. The GUI must be able to give

a detailed description of the network topology and of the network elements. The current GUI is designed to show the network elements, connections between these elements and mediation device, and the status of these entities. The GUI is basically used to maintain the configuration information that MDS/SAS requires for successful provisioning of the network. Instead the GUI should enable creation of connections between the network elements itself in order to establish links over the underlying network. The links created in the GUI should be propagated to the network level thus the router configuration would be automatic.

The other thing is that MDS/SAS should be able do is to maintain network level management information. Currently MDS/SAS expects that customer care systems maintain all the network level management information. However, with IP based services MDS/SAS should be able not just to maintain the topology of the underlying network but also the configurations of every individual network element in the service provider's network. MDS/SAS may also be required to maintain policy, security, accounting and SLA information depending on the service provider's requirements Thus, MDS/SAS requires a database for this kind of information. MDS/SAS has an internal relational database that is used to store request, configuration and logging information. However, a directory is a more convenient way to store network level information. An external LDAP directory should be added to the MDS/SAS or MDS/SAS could use existing network management information repositories.

It seems that MDS/SAS can only be used in situations where it acts as a traditional mediation device. One example situation is where MDS/SAS is used to update some particular network element. E.g. service provider has a network management system that is used to configure the core network and the routers within the network. However, there are a few network elements, such as Domain Name Server (DNS) or LDAP directory that require some configuration. MDS/SAS is used to interface to these network elements.

5 Lightweight Directory Access Protocol

5.1 Introduction

A directory is a specialised database. Current version of LDAP (version 3) stand-alone server is such a specialisation. Directories are optimised for read access and simple queries. A good example for a directory is Domain Name Server. It is updated rarely but read quite often by many clients [Wilc99]. LDAP introduces an extensible directory that can be used to store different types of information. It is accessible by an API and it is widely deployed. LDAP stand-alone server is robust, secure and scalable. [John98]

LDAP is an open industry standard for directory access. LDAP is widely accepted as the most important directory access method. Most of the major software vendors (Microsoft, Netscape, Sun, etc.) have implemented LDAP interfaces to their core products. Maybe the best example of the acceptance is the fact that Microsoft's Active Directory is compatible with LDAP, even thought the LDAP standard is nowadays largely driven by Netscape.

One can say that LDAP is one of the most important open standards in today's networking world. Previously there has not been any standardised access to directories that has enabled easy client development. This has lead to a situation where there are several applications, service and management platforms and other tools that use proprietary directories to store information such as usernames / passwords, email addresses, phone numbers, etc. The lack of centralised information repository has lead to a situation where network administrations must manage a number of different types of directories that have the same information. E.g. employee email address might be saved to company's password directory, global address book and department calendar. When the email address is changed the information must be update to all the three directories. Keeping this piece of information up-to-date and synchronised can be a daunting task, especially when there can be several administrators managing these separate systems. This is a challenging task and LDAP tries to answer this problem by enabling one centralised directory for all information. It has said to be the glue

that ties different directories together.

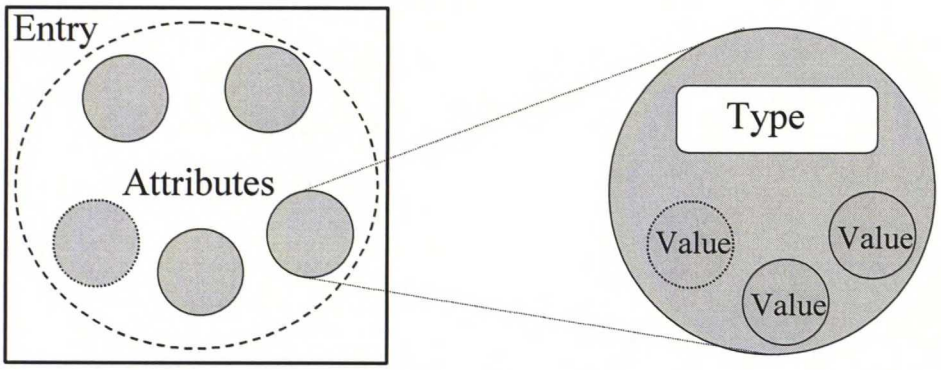
5.2 Evolution

LDAP was originally designed for accessing X.500 directories in the early 1990's. It was designed to replace the X.500 Directory Access Protocol (DAP), which was based on the OSI protocol stack. Those days the X.500 clients could not fit on the desktop computers so developers tried to find a less resource-intensive or lighter weight way to access X.500 directories, thus the name *Lightweight Directory Access Protocol*. The first LDAP standard was accepted by IETF in 1992. It used the more popular TCP/IP protocol stack that enabled easy X.500 client development. Since then LDAP has evolved beyond just an access protocol. In order to get away from the complexity of X.500 Directories, LDAP directories, i.e. LDAP stand-alone server was developed. The current version of LDAP is version 3. [Wilc98]

5.3 Concepts and Architecture

5.3.1 Entry

Entry is the basic informational entity stored in a LDAP directory. An entry represents an object that contains a collection of attributes. Attributes contain the information about the object. Every attribute has a type and one or more values. Syntax of the type defines what kind of information can be stored in a particular attribute and also how these values behave during LDAP operations. Also the number of values and the size of a value can be restricted. Figure 5-1 shows an example entry with several attributes. Figure also lists some of the syntaxes that are defined for LDAP. [RFC2251]



Attribute / Alias (Type)	Syntax	Description	Example
Telephone Number / telephoneNumber	tel	telephone numbers	050-5548959, 09-4775667

Syntax	Description
tel	Numbers are treated as text, but all blanks and dashes are ignored.
bin	Binary Information (e.g. JPEG image)
dn	Distinguished name

Figure 5-1: Entry, Attribute, Value and Syntax

5.3.2 Schema

Schema is used to define the types of objects that can be stored into the directory. Schema is also used to define the types of operations that are allowed on the objects and how these functions are executed related to these objects. It also defines the attributes of particular entry and whether they are optional. An entry cannot be stored before all the required attributes of this entry are saved. One of the most powerful features of the LDAP standard is the possibility to derive objects from other objects. Organisations can define their own attributers, such as company key identification number, for example by sub-classing standardised *organizationObject* with the additional attributes. Every entry has a compulsory attribute that defines the object class (*objectClass*). The value of the attribute *objectClass* is a list of two or more schema names that define the types of objects that the entry represents. Many common schemas are standardised even though every LDAP

server can define their own schema. [RFC2251]

5.3.3 Directory Information Tree

DIT is made up of entries. A Distinguished Name (DN) identifies an entry in the DIT. DN must be unique for every entry in the directory. DN is made up of a sequence of Relative Distinguished Names (RDNs) that are separated by commas. RDN is made of a collection of attributes and their values in the entry. It is difficult to define RDN exactly since there are no rules for creating a RND. Usually RND is considered to be the left most part of the DN. One or more LDAP servers can maintain one DIT. There might be e.g. one server responsible for one department of an organisation. Referrals are used to link these separate LDAP servers together to form a distributed directory for holding the whole DIT. A referral is an attribute which value is the Uniform Resource Locator (URL) of the entry in the other LDAP server. LDAP client does not have to be aware that there are several LDAP servers because LDAP Application Programming Interface (API) can automatically execute the LDAP operation based on the referrals. [Figure 5-2] [Wilc99]

Figure 5-2 depicts a situation where the DIT is separated between two LDAP servers. The branch office located in Kuala Lumpur uses server 1. The other server is located in Helsinki and it acts as the master server. Both servers store local research and development information but only the LDAP server in Helsinki is responsible for the customer information. Referrals are used to link these two servers together. If a client that is connected to the server 2 makes a query using suffix loc=kl,o=comptel.com then a referral is returned in the response. Thus, the information is retrieved from the server located in Kuala Lumpur.

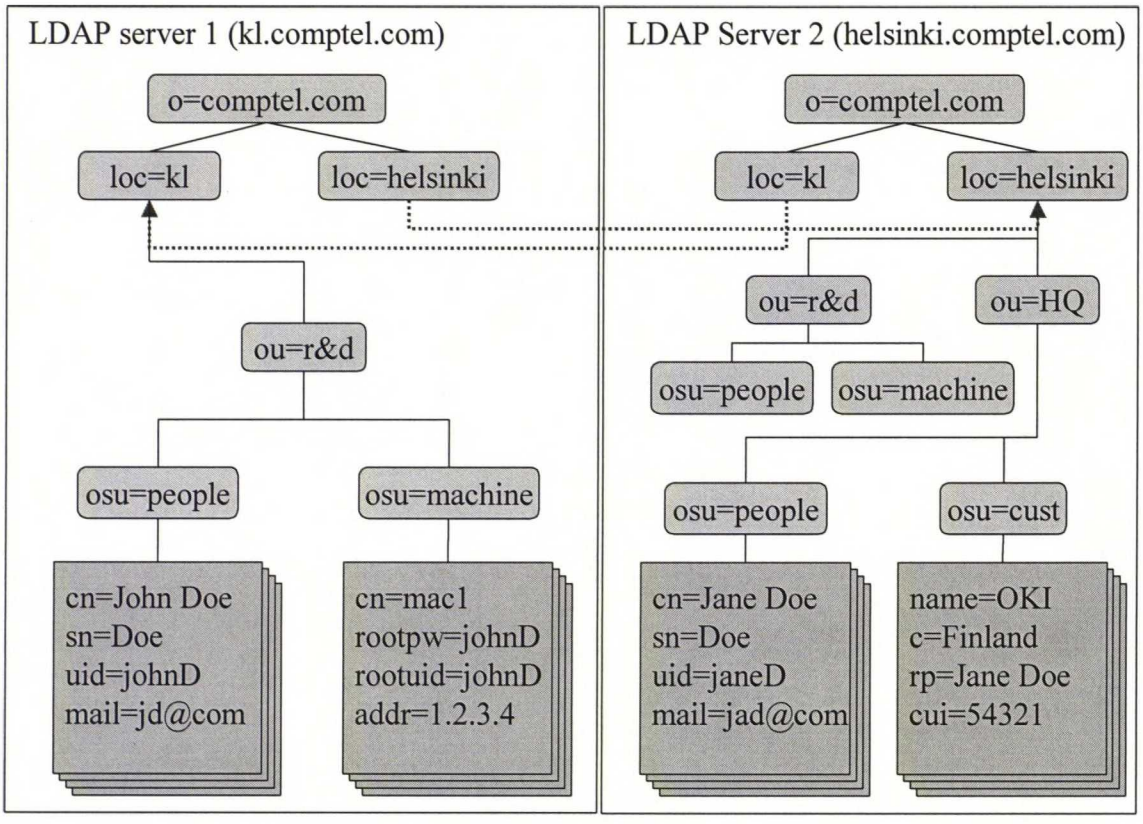


Figure 5-2: LDIT and Referral

5.3.4 LDAP Data Interchange Format

LDAP Data Interchange Format (LDIF) is a text-based format that describes the directory information or modifications made to the directory information. LDIF has been developed to enable mass import and export of information to and from LDAP directories. It is mostly used when directory information is transferred from one LDAP server to another or when there is a need to apply a set of changes to a LDAP directory. Example of a LDIF file is shown in figure 5-3. It also describes the basic format of a LDIF file. LDIF file is a collection of records separated by line separators. A record consists of a sequence of lines describing a directory entry, or a sequence of lines describing a set of changes to a directory entry. Even though LDIF is a powerful tool it can also be a security risk. Since LDIF file is a normal file one must take care not to allow it sensitive context be read by anyone. LDIF also does not provide any method for authenticating the LDIF file so

one must be careful when using a LDIF file received from an external source. For example the LDIF reference directive can be used to include undesirable information to the directory. [RFC2849]

```
version: 1
# This is an example LDIF files that holds
# two directory entries.

dn: cn=John Doe, ou=r&d, o=comptel.com
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: John Doe
cn: John V. Doe
sn: Doe
uid: jd
telephonenumber: +1235678
description: LDAP Expert.

dn: cn=Jane Doe, ou=HQ, o=comptel.com
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: Jane Doe
sn: Doe
telephonenumber: +87654321
# Reference to an external file
jpegphoto:< file:///home/JaneDoe.jpg
```

```
[<version number>]
# Basic LDIF form specifying a LDAP entry
dn: <distinguished Name>
objectClass: <object class>
objectClass: <object class>
...
<attribute type>[;language tag]:<attributevalue>
<attribute type>[;language tag]:<attributevalue>
...
```

```
version: 1
# LDIF file with two change records
#add new entry
dn: cn=Jack Doe, ou=HQ, o=comptel.com
changetype: add
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: Jack Doe
telephonenumber: +1243456578

# Delete an existing entry
dn: cn=Jane Doe, ou=HQ, o=comptel.com
changetype: delete
```

Figure 5-3: LDIF file examples

5.3.5 LDAP API

LDAP protocol and architecture itself do not describe a LDAP API. However, there is an informational RFC [RFC1823] that defines a C language LDAP API. This API has been used to develop a de facto standard LDAP API that is supported by most of the major LDAP vendors. Nowadays there is also LDAP APIs for C++, Perl and Java languages. The philosophy of the LDAP API is to keep things simple and this means that LDAP client

development is relatively easy. LDAP vendors have provided several free LDAP API Software Development Kits (SDKs) for the developer community. Client application generally uses LDAP API in four steps [RFC1823]:

1. Open a connection to a LDAP server. LDAP server can handle multiple connections at the same time.

2. Authenticate to the LDAP server. [RFC2829] defines the authentication methods. The security aspects of LDAP are discussed in more detail in the following chapters.

3. Execute some LDAP operations such as add, delete, etc. Operations can be executed synchronously or asynchronously.

4. Close the connection.

The referral functionality is implemented in the LDAP API. Client can define that the referrals are followed automatically. When the response to client's LDAP request is a referral it can be followed automatically so that neither LDAP server nor the clients are required to implement referral functionality. This makes the client development easier and also improves the LDAP server performance. [John98]

5.3.6 LDAP Operations

As mentioned above the philosophy of LDAP API is to keep simple things simple. This can be seen on the LDAP operations that client uses. LDAP defines operation for modifying and accessing directory entries. Below is a description of the different operations that LDAP supports. LDAP operations can be divided into four categories: query, update, authentication and controls and extended operations [John98]. Only operations and functionality related to LDAP version 3 will be discussed if not otherwise

mentioned. [RFC2251]

Query

Search is the most commonly used operation in LDAP. As mentioned earlier directories in general are read more often than updated, so they are optimised for reading. Search is the most flexible and sophisticated operation that LDAP offers. There are several mandatory search parameters (some of these parameters can be optional when using LDAP API SDKs):

1. Base. This is a distinguished name that defines the starting point for the search. It is called the base object and it must be a node in the DIT.

2. Scope. Scope defines the depth of the search from the base object. Scope can be one of three values: *baseObject, singleLevel, wholeSubtree.* baseObject means that only the base object is examined. singleLevel means that the base object is not examined but only the immediate children. wholeSubtree specifies that the whole sub-tree, including the base object, is examined.

3. Search filter. Search filter specifies the criteria that an entry must match so that the information in the entry is returned. Filter uses the form: (attribute operator value). Attribute can be any LDAP attribute even one that is not specified in the LDAP schema. Operators are listed in the table 5-1 [John98]. Boolean operators apply to a whole filter instead of just filter values so they can be used to create more complex search filters [table 5-2] [John98]. Value can be any string value and wildcards (*) are also accepted. The case sensitivity is defined by the attribute (defined in directory's schema), e.g. attribute *cn* in case insensitive but attribute *labeledURL* is case sensitive. [RFC2254] Table 5-3 illustrates some example search filters.

4. Attribute types only. This parameter can be used to define whether only the

attribute types or both attribute types and values are returned.

5. Alias Dereferencing. – Indicates how alias objects are to be handled.

6. Size and time limits. Some times there is a need to limit the search. When searching through a large sub-tree using a general search filter the operation can consume huge amounts of resources. In such a case it is proper to use either time or size restrictions. Time limit limits the total search time (in seconds). The size limit on the other hand limits the number of entries returned as a result of the search. LDAP server can override these values by defining stricter limits.

7. Attributes to return: This parameter can be used to define a list of attributes that are to be returned from each entry that matches the search filter. All user attributes are returned if no attribute list is defined.

As one can see the search operation is very flexible. One can use general search filter (e.g. list all employees) and return only the required attributes from a particular point in the DIT. Or one can use very specific filter that only returns one required value, such as user email address.

Table 5-1: Search filter operators

Operator	Description
=	Returns entries whose attribute is equal to value
>=	Returns entries whose attribute is greater than or equal to value
<=	Returns entries whose attribute is less than or equal to value
=*	Returns entries that have a value set for that attribute
~=	Returns entries whose attribute value approximately matches the specified value.

Table 5-2: Boolean operators

Operator	Description
&	Returns entries matching all specified filter criteria
|	Returns entries matching one or more of the filter criteria
!	Returns entries for which the filter is not true. Note! This operator can be used only with one filter, i.e. (!(filter1)) = ok, (!(filter1)(filter2)) = error

Table 5-3: Search filter examples

Filter	Description
(cn=John Doe)	Search for an entry with common name of John Doe
(cn=*Doe)	Search for all entries with common name attribute ending to Doe
(sn<=Doe)	Search for all entries that have surname less than or equal to Doe. Such names are Dod, Doc, etc.
(|(ou=r&d)(ou=HQ))	Search for all entries that have organisational units of r&d or HQ
(&((|(sn=Doe)(sn=De))(ou=r&d))	Search for all entries that have surnames of Doe or Deo and have an organisational unit of r&d

Compare is also quite commonly used operation. It is very useful for example when using LDAP server as a password directory. LDAP server can be configured so that the user password attribute can be compared but not read. Compare operation returns TRUE if the entry has compared value.

Update

There are four operations that are capable of modifying the content of a LDAP directory: Add, delete, modify and modify DN. All these operations are only request to a LDAP

server from the client's point of view. Client must have adequate user rights for the request to be successful.

Add operation adds a new entry to the LDAP directory. A unique distinguished name must be provided to a new entry. Also the distinguished values, which form the RND, must be included in the request. Mandatory object class attributes defines those attributes that must be also included (e.g. objectClass=Person -> mandatory attributes are both cn and sn).

Delete operation is the most straightforward operation. It can be used to delete a leaf entry and the only required parameter is the distinguished name.

Modify operation is used to modify an existing entry. Modify can be used to add new attributes, modify existing attributes or deleting existing attributes. Distinguished name is the only mandatory parameter.

Modify DN is used to change the leftmost (least significant) component of the name of an entry in the directory, or to move a sub-tree of entries to a new location in the DIT. It is, however, not possible to move entries across server boundaries. There are four parameters that can be used in this operation: entry, newrdn, deleteoldrdn and newSuperior. entry is the distinguished name of the entry that is to be changed. newrdn is the RDN that will form the leftmost component of the new name of the entry. deleteoldrdn is a boolean parameter that is used to control how the old RDN attribute values are handled. They are either deleted from the entry or retained as attributes of the entry. newSuperior is, if present, the distinguished name of the entry which becomes the immediate superior of the existing entry.

Authentication

There are three authentication operations: bind, unbind and abandon. These methods are used to manage the client-server session. A LDAP session can have several levels of security from an insecure anonymous session and password protected authenticated session

to a highly secure, encrypted session where Simple Authentication and Security Layer (SASL) mechanism is used provide security services.

Bind operation is used to exchange authentication information between client and server. Bind operation has three parameters: version, name and authentication. Version parameter defines the version of the LDAP protocol that should be used in particular session. Name parameter specifies the name of the directory object that the client wishes to bind as. This field can be null if anonymous bind is used or if some other lower layer authentication mechanism is used, like SASL. Authentication parameter can be used to carry the information that is used to authenticate the name parameter.

Unbind operation terminates the protocol session.

Abandon operation can be used to abandon an outstanding operation. E.g. when a search is taking too much time, the client can send the server an abandon request and the server kills the operation. The correct operation is identified by a MessageID parameter that the client has received when sending the operation request.

Controls and Extended Operations

Controls and Extended operations are added in order to make sure that LDAP version 3 is flexible enough for future needs. These operations can be used to extend the functionality LDAP without making any changes to the protocol itself.

Controls are used to modify the behaviour of any existing LDAP operation. Controls are new parameters in overloaded LDAP API functions. Standard LDAP controls have been proposed in various RFCs, e.g. [RFC2891]. The LDAP server must support the controls that the client is using otherwise the LDAP server ignores them.

Extended operations can be used to define new operations. For example [RFC3062] describes an extended operation intended to allow directory clients to update user

passwords. Standard LDAP update operation does not allow a user to update his/her password when a DN does not represent the user, does not have an entry or when the password used by the server is not stored as an attribute of an entry.

5.3.7 Security

Security is one of the most important aspects when talking about directories or data storages in general. LDAP servers often hold highly sensitive information, like passwords and credit card numbers. Therefore additional effort must be taken in order to make the usage of LDAP directories secure.

[RFC2829] identifies following security threats to a LDAP directory from hostile clients: Unauthorised access to data via data-fetching operations, unauthorised modification of data and LDAP server configuration and unauthorised or excessive use of resources e.g. Denial of Service (DoS) attack. Man-in-the-middle threats include monitoring others' access, which can lead to either unauthorised access to reusable client authentication information or unauthorised access to data. Spoofing of directory, i.e. tricking a client into believing that information came from the directory when in fact it did not. Spoofing can be done either by modifying data in transit or misdirecting the client's connection.

[RFC2829] recommends following security mechanisms to protect the LDAP protocol suite: Client/server authentication by using the SASL mechanism set and the Transport Layer Security (TLS) credentials exchange mechanism. Requestor's authenticated identity based access control for client authorisation. TLS protocol or data-integrity SASL mechanisms for protecting the integrity of the data. Protection against snooping using the TLS protocol or data-encrypting SASL mechanisms. Resource limitation by using administrative limits on LDAP server (i.e. time and size limits).

5.4 Conclusions

LDAP provides a single repository for all kind of directory data. It has been widely

deployed and today almost every major software and hardware vendor is shipping products with LDAP interfaces. As LDAP is an open standard everyone has the ability to interface with LDAP directories. LDAP also provides the security, extendibility, scalability and performance that today's network solutions require. LDAP is considered to be one of the most important interfaces that MDS/SAS must be able to support. One good example of the potential of LDAP is the Directory Enabled Networks (DEN) concept described in next chapter.

5.5 Directory Enabled Networks

Cisco and Microsoft first introduced the DEN initiative in 1997. It is open, industry-wide concept that provides building blocks for intelligent networks by mapping users to network services, and mapping business criteria to the delivery or network services. DEN will enable easier service creation, provisioning and management on large-scale distributed networks. DEN specification relies on LDAP as a core data access protocol. LDAP is used to access, manage, and manipulate directory information. [DMTF].

Networks today are complex and the management of these networks is time-consuming and error prone. One of the major problems with this situation is that service creation is hard, costly and it is difficult to activate these services for the customers. Current directory service technology is not designed to meet the changing needs of today's network applications. The idea behind DEN is to transform the current data-storage type of directory into authoritative, distributed, intelligent information repository for all service related data. This is a major shift towards an idea where a directory is the foundation for an intelligent infrastructure. DEN schema and information model augment existing network services and associated protocols. These include e.g. DNS, Dynamic Host Configuration Protocol (DHCP) and RADIUS. For example DEN concept can be used to create, maintain and provision dynamic services over a large-scale and complex networks. One example is a 24/7 video-on-demand service for a home user. These kinds of services can be managed more easily when management information about the users, networks and services is

available in a centralised information repository. [Cisc99]

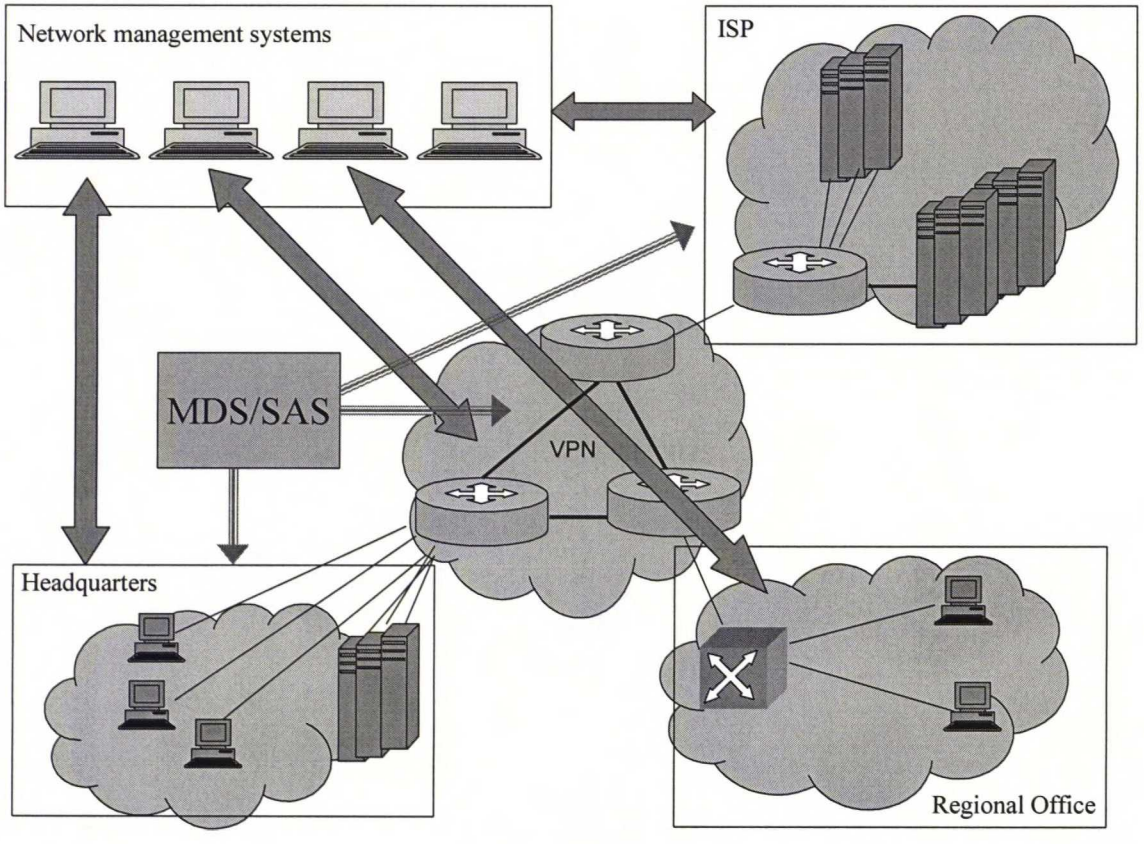


Figure 5-4: Network without centralised management

Figures 5-4 and 5-5 shows the difference between a common network and DEN from the system management's point of view. One can see that DEN defines system as a network or network-wide end-to-end service or application. Where as today's networks consider system to be a network element in a management domain. All the functions, such as management, configuration and security, are network element based. In addition to the problems mentioned earlier there is also a scalability problem with today's networks. These islands of systems are more or less non-interoperable because there are not any widely deployment management standards. DEN enables a scalable network solution that is based on open standards. Because the service information is centralised there is basically one point of provisioning in the system. Thus integration of MDS/SAS into a DEN is not

as challenging as it is to manage today's networks.

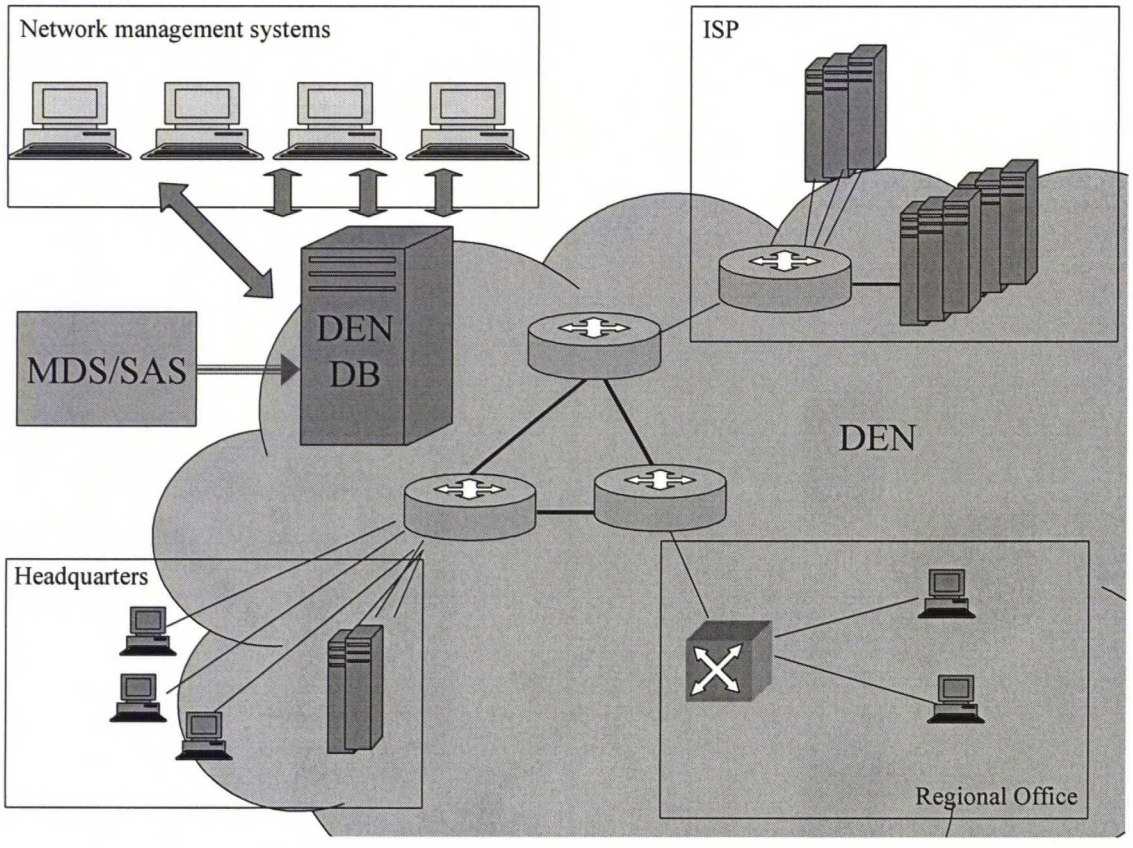


Figure 5-5: Directory Enabled Network

5.5.1 DEN Specification

DEN specification is two-fold: it defines both a standard schema for storing persistent state and an information model for describing the relationships among objects representing users, applications, network elements and network services. DEN specification relies on existing network management protocols, such as SNMP, in network element interfaces and uses LDAP as the access protocol to the directory.

DEN concept creates an environment in which the directory information can be used by several applications in order to provide enhanced functionality and services. An

information model is an abstraction of knowledge and applications interact using it. The information about users, applications, networks, and how they interact is structured into multiple domains. LDAP provides schemas that define users and applications. DEN specification defines a schema that defines network elements and services and how they interact with applications, users, and other services. The information model consists of three parts [Judd98]:

1. Eight base classes that form the basic framework

2. An extensible schema based on inheritance and aggregation for modelling application-specific properties and information

3. Simple mechanisms for establishing relationships among object instances.

In all there are eight abstract base classes. These are Person and Device which are augmented X.500 classes, and Application, Protocol, Media, Profile, Policy, and Service. Person and Device classes are used to describe and control the interaction among users, applications, network elements and services. The other six abstract classes describe network element and service definitions. Application specific information can be added by deriving subclasses from each abstract base class in the framework. This information model is a robust object-oriented model, which enables applications that have completely different purposes but which are operating on common objects to exchange information and knowledge about those objects. Interoperability among implementations is maintained, because all implementations are refinements of a common base.

Because network elements and services are complex objects and exist in a constantly changing environment it is insufficient to describe them using only class hierarchies. Therefore network elements and services must be modelled. There are three aspects to modelling complex systems [Judd98]:

1. Object Models are used to describe the static structure (data, attributes,

operations, and relationship to other components) of the system.

2. Dynamic Models are used to describe the temporal relationships (i.e., the behaviour) between the different system components, and how each component is controlled as a function of time (e.g., assigning particular events or changes in state)

3. Functional Models are used to describe relationships among values, and how functions, mappings, and constraints are used within the system to determine the final value of an item (e.g., the data flow).

These models do not exclude each other, on the contrary. It is required that all three models are considered together if network elements and services are complex.

The DEN concept requires a common information repository for holding a common model describing the structure, behaviour, and operation of a network element or service. This common model is required for enabling the exchange of information between network elements. Model information describes features, available services, current configuration information, and supported protocols and APIs of the network elements. There are four different types of information that are necessary to model the structural information of network elements and services [Judd98].

1. Intrinsic. This category represents information that is essential for representing a particular element or service. There are two types of intrinsic data:

    a. Data that never changes (e.g., the MAC address or the serial number of the device)

    b. Data that changes infrequently (e.g., the device name or its IP address). The defining characteristic of intrinsic data is that it is used to uniquely identify

that device or service.

2. Configurable. This category represents information that controls the operation of a device, or helps determine how that device or service operates. This data remains static until the configuration of the device or service is explicitly changed (e.g., a new interface processor is put into a router). Furthermore, the values that data belonging to this category can take for devices are always chosen from a pre-defined list. A defining characteristic of configurable data is that it is used to define how the device or service will function.

3. Operational. This category represents information that controls how a device or service interacts with its surrounding environment. This information is derived from the operational environment of the device or service, and changes as a result of operating conditions such as network load. A differentiating characteristic of operational data is that it summarises how the device or service functions without any manual intervention (e.g., routing tables are automatically updated by routing protocols as the conditions change).

4. Contextual. This category represents information defining how the device or service relates to other components in a larger, network-wide context. A differentiating characteristic of this type of information is that it must be used with lower-level device information, especially intrinsic and operational data, in order to be meaningful.

5.5.2 Base Schema

DEN schema consists of the abstract base classes mentioned earlier. Common Information Model (CIM), DEN and X.500 define these base classes. All other classes are derived from these classes. The base schema is shown in figure 5-6. It also shows one derived class (i.e. Protocol). The work to create new standard derived class is underway in order to enable

DEN implementations. [Judd98]

Top is an X.500 base class that forms the starting point for all other classes. Other three X.500 base classes are Device, which is used to represent any physical device, Person for representing a human user, and Application-Process base class for services and applications. The five CIM base classes are described in CIM specification. Managed-System-Element forms the starting point for all physical devices, software components, and other system components that can be modelled. Location defines a geographic location where physical devices can be installed. Protocol class is used to represent different protocols and Service describes the services that are offered on the network. The final CIM base class Application is for applications that consume network resources. The base classes that are especially designed for DEN are Profile, which encapsulates information governing the characteristics and needs of a specific principal. Policy encapsulates information governing the use of network resources in a particular context and the way that different network resources interact with each other. Network-Media base class for defining the way that network elements use different media to communicate with each other and with other system components. And finally Linked-Container, which is a container class that implements a forward link, allowing containers to be arranged as a forward linked list. [Judd98]

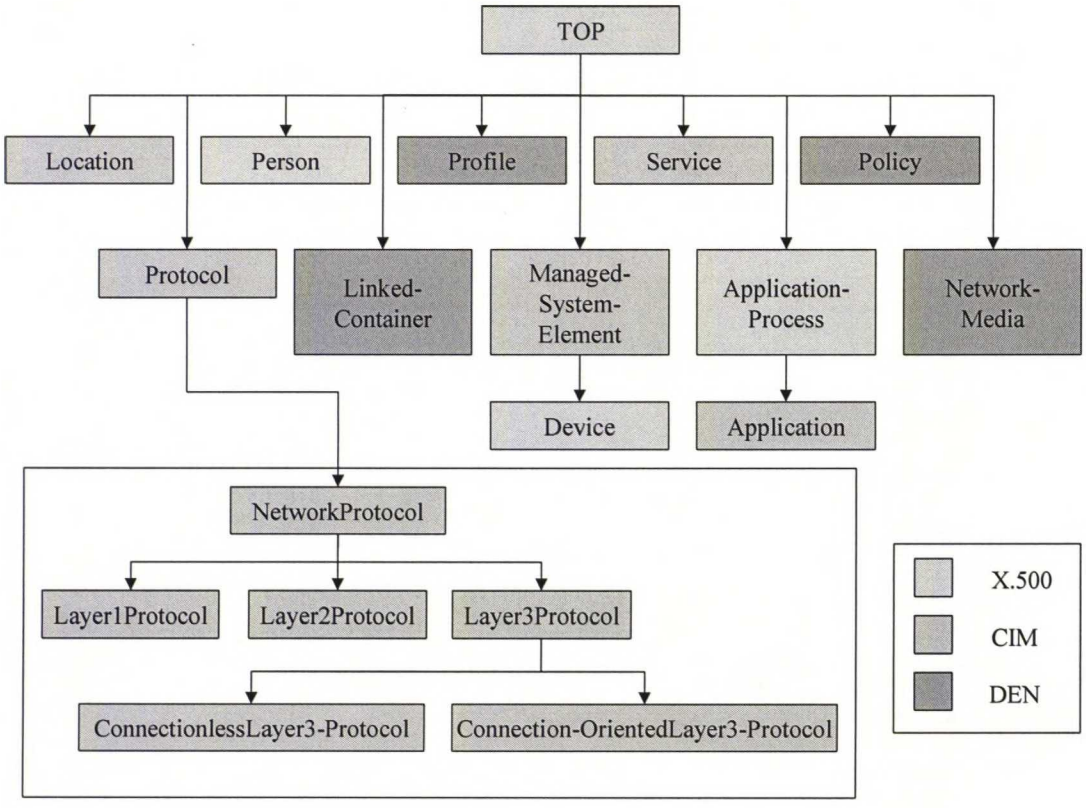


Figure 5-6: Functional structure of the DEN base classes

6 Network Element Interface Implementation

6.1 System Overview

6.1.1 Application Domain

Java Macro Server operates between MDS/SAS core and network elements. It carries out tasks performing service provisioning in response to requests from MDS/SAS. LDAP network element interface module uses JAVA-API. It extends MDS/SAS core product by enabling easy handling of Java Interfaces towards network elements. [Java01]

6.1.2 External Connectivity

Java Macro Server communicates with MDS/SAS core component via Local Command Executor (LEX). LEX is a MDS/SAS core component that receives executable tasks from other components and transfers these tasks to the NE interface layer. Java Macro Server uses standard I/O to transfer information to and from LEX (figure 6-1). Java Macro Server does not make any assumptions about the NE interface. It can be used with several different technologies, such as LDAP.

6.1.3 Hardware Platform

Java Macro Server does not make any assumptions about the underlying hardware.

6.1.4 Software Platform

The development software version of Java is 1.1.x, which is compatible with MDS/SAS 4 and is delivered within the MDS/SAS 4 package. The development guideline is to use the newest Java version that is compatible with MDS/SAS 4 and it will be updated to a newer

version if needed. Related software includes Comptel standard libraries. Java was chosen as the programming language due to its wide usage and easy portability.

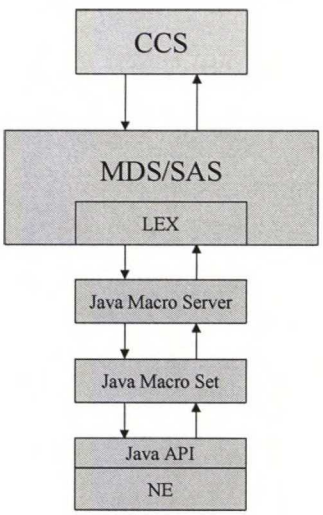


Figure 6-1: Java Macro Server and Macro Set

6.2 Architectural Description

6.2.1 Design Philosophy

Due to the evolving requirements and versatility of LDAP directory implementations, main design considerations include modifiability, extensibility and maintainability. One major design principle of the LDAP interface is to get a generic LDAP interface module that can be used from one customer implementation to another with only minor modifications. Thus, to have a generic software module that does not require additional coding in different situations. These criteria are achieved with runtime parameterisation through user-defined templates or by LDAP specific MDS/SAS request parameters. This document will introduce both provisioning approaches.

6.2.2 Database Architecture

Java Macro Server does not use any separate DBMS. A Global Resource Configuration (GRC) file is used for software parameterisation and log entries are written to normal flat files.

6.2.3 Software Architecture

The class diagram of Java Macro Server is depicted in figure 6-2. [Java01]

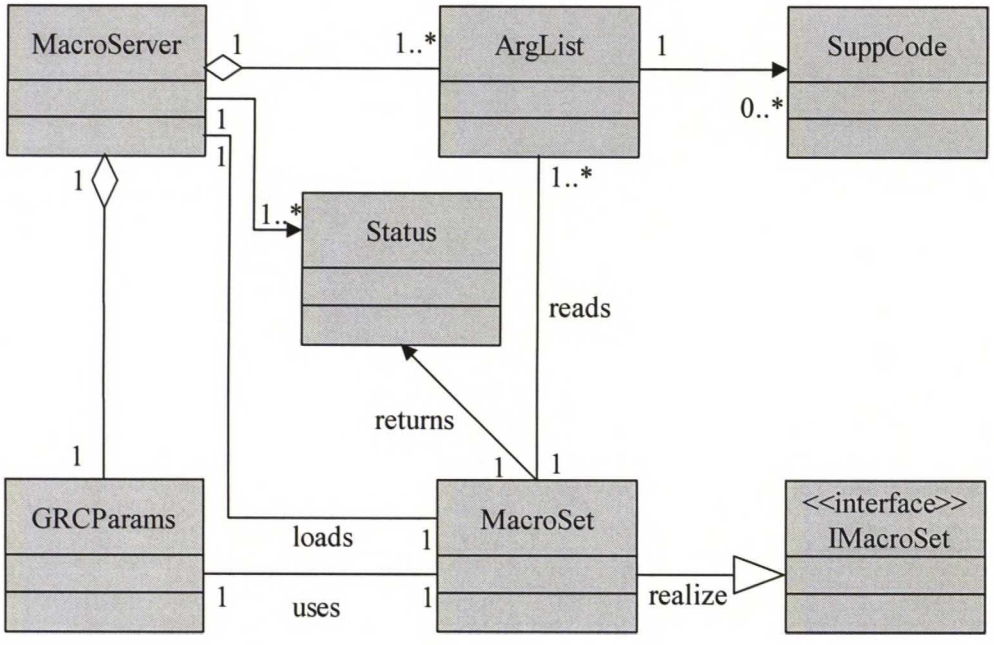


Figure 6-2: Class Diagram

The main Java Macro Server class, called MacroServer, handles the main macro server functionality. MacroServer class methods are used to initialise the Java Macro Server. Initialisation includes the reading of GRC parameters and other configuration files. MacroServer main procedure reads commands from LEX and calls methods accordingly.

Class Status is responsible for maintaining both the status of the network element

connection and the status of the execution of task returned by macro set. Macro set class must return task and connection statuses as a return value to the Macro Server.

Class SuppCode handles supplementary service codes.

Service request that is sent to Macro Server by LEX can contain several request parameters and request parameter values. Class Arglist is used to create an argument list from the request. This argument list is then passed to the macro set. The argument list includes request parameters originated both from CCS and LEX. CCS parameters are in upper case and LEX parameters in lower case.

Class GRCParams is used to handle the GRC file. It has methods for retrieving GRC parameters.

Class MacroSet realizes IMacroSet interface and implements the methods of IMacroSet interface.

A macro set inherits the methods of class MacroSet. Macro set contains the main functionality related to the LDAP provisioning. The methods of this class are responsible for enabling the LDAP operations. The operations are implemented in this class. Also the template functionality is implemented here. The service request is examined and the mode of execution is decided based on the request parameters.

6.3 Procedure Descriptions

This chapter defines the request parameter names to be used by MDS/SAS with LDAP. The template grammars are described in the next chapter. The supported service request types are:

- **Create:** This request type is used for adding one or more entries.

- **Modify:** This request type is used for modifying the attributes of one or more entries.

- **Delete:** This request type is used for deleting one or more entries.

- **Display:** This request type is used for querying information from the LDAP directory.

All of these service requests can be executed in one of two ways. Either using LDAP specific MDS/SAS request parameters of by using templates. By using templates the number of request parameters can be reduced. This simplifies the service request and this approach is required by some customer care systems.

Macro set examines the service request and decides whether to use template. Macro set makes the decision based on the request parameters. The LDAPFUNC- request parameter is used to specify the correct template. A template corresponds to one request so that only one template can be executed at a time. Template can contain several entries so that all the required objects can be created to the LDAP directory with one request. If the LDAP specific parameter approach is used then the LDAP specific parameters must be marked so that the parameters can be separated from other MDS/SAS request parameters.

All of the service request examples presented in this chapter are based on the customer case described in chapter 4.

6.3.1 Template Approach

When using a template approach only those parameters that are entry specific must be sent in the MDS/SAS service request. For example if one creates several employee entries to a same department then those attribute values that are same for all the employees can be specified in the template. E.g. company name and other organisational information.

The template that is executed in the "create"- example is shown in appendix 1 Template approach works fine with LDAP operations. Especially with create and search operations since the number of parameters can be quite large. It is easy and fast to create customer specific implementations by creating a generic LDAP macro set and using customer specific templates. Template holds all the parameters that the LDAP schema enables to a particular entry. Those template parameters that are not sent in the request are discarded. Some of the attributes are market in the template as "Not used". These are basically hard coded values that are used just to indicate that these attributes do not receive values. Only on the mandatory attributes needs to be added to the template. The only reason to do this is to make it clear that the attributes do not require values by saving the information into the directory.

Create

Create method is used to add new entries to the LDAP directory. Both a unique distinguished name and all of the distinguished values, which form the RND must be provided for create method. All the mandatory object class attributes must be specified also; otherwise the LDAP service will reject the request.

Create method must be able to add three entries into the LDAP directory, one for each branch (customer, device and connection). The most convenient way create these entries is to use template approach, because it can be used to specify all the three entries. Thus, one service request corresponds to three network layer tasks. Those attributes that are same to different entries (e.g. vpnName) are sent only once.

Table 6-1: Request Parameters for Create

Request Parameter	Example
LDAPFUNC	Create_VPN_customer.ldap
DN_CUSTOMER	vpnName=VPN_A, ou=Comptel, dc=GoldenConnection, dc=com
DN_DEVICE	vpnName=VPN_A, cn=CE-G, dc=GoldenConnection, dc=com

Request Parameter	Example
DN_CONNECTION	vpnName=VPN_A, subnetNumber =20.8.8.8, dc=GoldenConnection, dc=com
OU	Comptel
FAX	+3589123456
LOCATION	Helsinki
POSTALADDRESS	Ruoholahdenkatu 99
POSTALCODE	00100
POSTOFFICEBOX	PO123
STREET	Ruoholahdenkatu 99
TN	+35897654321
PASSWORD	password
VPN_NAME	VPN_A
VPN_SITE_NAME	Helsinki_1
ACTIVATION_DATE	30082001
SLA_URL	http://intra.sla.com/VPNCustomer/Comptel/
CONTACT_PERSON	Jukka Halonen
CE_NAME	CE-I
SERIALNUMBER	S/N: 1234556778
DEVICEID	VPN_A-Helsinki_1-20.0.0.9
PEERID	PE-F
IPADDRESS	20.0.0.9/32
VENDOR	Cisco
MODEL	7240
OS	12.7(T)
MACADDRESS	C0A86400:40F9001859
SUBNET	20.8.8.8/32
SUBNETMASK	255.255.255.0

Response

Task execution successful

No specific LDAP response.

Task execution erroneous

A LDAP server's exception message of a LDAP macro server /macro set error message.

Modify

This operation is used to modify an existing entry. Modify can be used to add new attributes, modify existing attributes or deleting existing attributes. Distinguished name is the only mandatory parameter

The template approach works fine with modify functionality. Modify functionality is implemented so that modify template holds all the possible attributes of entries that must be modified. Only those attributes are updated that are specified in the service request. Those attributes that are defined in the template parameters and do not receive a request parameter are discarded. The functionality is quite similar to the create functionality from the point of view of MDS/SAS.

Response

Task execution successful

No specific LDAP response.

Task execution erroneous

A LDAP server's exception message of a LDAP macro server /macro set error message.

Delete

Delete functionality is straightforward. It can be used to delete a leaf entry and the only

required parameter is the distinguished name.

There is only one clear reason to use template with delete- functionality; template can be used to delete several entries. If all the other methods use template then it could be wise that delete is also executed using a template.

Response

Task execution successful

No specific LDAP response.

Task execution erroneous

A LDAP server's exception message of a LDAP macro server /macro set error message.

Search

Search is the most commonly used operation because directories are generally read more often than updated. Search is the most flexible and sophisticated operation that LDAP offers. There are several mandatory search parameters.

Search is by far the most challenging method. Great emphasis must be put to specify this functionality because LDAP search and the response generation functionalities are so flexible. The role of MDS/SAS is one thing we should consider. Should MDS/SAS be able to execute large-scale searches where the response can contains several attributes from multiple entries, or is the emphasis on searches where result of the response is limited to attributes of a single entry (or even single attribute, say password). It seems that MDS/SAS is used mainly to maintain the directory so that other client (such as NMS) can read from it. However, MDS/SAS must be able to search LDAP directory, thus this functionality must also be supported.

The problem with search is not related to the complexity of the request but to the complexity of the response. The responses from LDAP can contain a number of different attributes from several entries. We can restrict the search method so that MDS/SAS only executes searches where the response is quite small. There are other tools for large-scale search of the directory. The main task for MDS/SAS is to read just a single or a few attributes.

The template used in search procedure is different from the templates used in other procedures due to the nature of the operation. The request parameters that are required by search functionality are mentioned in Table 6-2. The main advantage of a template when used with a search method is that it can define some static parameters, such as search restrictions. An example template is described in Appendix 2.

Table 6-2: Request Parameters for Search

Request Parameter	Mandatory	Example
LDAPFUNC	M	Search_customer_device.ldap
DN	M	vpnName=VPN_A, ou=device, dc=GoldenConnection, dc=com
SCOPE	M	SCOPE_BASE
FILTER	M	FILTER=(deviceId=VPN_A*)

Response

Task execution successful

The response of the LDAP search can be complicated. The response can contain several attribute values from several entries. The biggest problem is to find a way to modify the response so that it can be sent back to the CCS through MDS/SAS elegantly. Below are a few examples.

There are no problems if the search is limited only for a single or a few attribute values in a single entry. Different attributes are distinguished using parentheses.

LDAPDATA((cn= VPN_A-Helsinki_1-20.0.0.9)(vpnName= VPN_A))

But if the response contains attribute values from several entries e.g. (100 entries) there can be problems with sending the huge amount of data through MDS/SAS. There is also the problem of differentiating the different entries from each other. Here we use brackets.

LDAPDATA([(cn= VPN_A-Helsinki_1-20.0.0.9)(vpnName= VPN_A)][(cn= VPN_B-Helsinki_2-20.0.0.3)(vpnName= VPN_B)])

The problem is that MDS/SAS is not designed to be used in situations where large responses are required. The MDS/SAS core can support efficiently only small responses, such as exceptions thrown by the NE or basic customer information.

The other problem with response in addition to the size is the format of the response. The LDAP information is stored in entries thus MDS/SAS should be able to format the response naturally based on an entry. This solution uses brackets and parenthesis to separate the different entries and attributes from one another. However, this is not a elegant way, even though parsers do not have any trouble handling this type of data. It could be wise to use e.g. XML-format in the response. MDS/SAS could be modified to use XML internally so that it could better adapt to situations like this.

Task execution erroneous

A LDAP server's exception message of a LDAP macro server /macro set error message.

6.3.2 LDAP Specific Parameter Approach

The other method for execution of LDAP request is to use the LDAP specific parameter approach. Every LDAP specific parameter must be identified when using the LDAP specific parameter approach. I.e. LDAP NEI module must be able to separate LDAP parameters from other MDS/SAS request parameters. This is achieved by using a

prefix (e.g. cn -> LDAP_cn). A macro set parses the LDAP specific parameters and removes the prefix. Every LDAP specific parameter is used as such, thus these parameters must be sent in the correct LDAP format. All the parameters (LDAP attributes) that LDAP requires must be defined in the request. The prefix is defined in the GRC file.

Create

Three service requests must be created in order to create all the three entries, because LDAP NEI module is not able to separate the parameters of the different entries. Only the attributes of the Customer- branch are listed in table 6-3. Other cases follow the same notation. When using this approach we can create a generic LDAP macro set that is able to create an entry to any LDAP directory without any customer specific modifications. However all of the CCSs may not be able to send the requests in correct format. The LDAP interface module acts only as a gateway to the LDAP directory and does not really make the provisioning any easier for the CCS.

Table 6-3: Request Parameters for Create

Request Parameter	Mandatory	Example
DN	Yes	vpnName=VPN_A, ou=Comptel, dc=GoldenConnection, dc=com
LDAP_objectClass	Yes	top,organizationalUnit,organizationalUnitCust
LDAP_ou	Yes	Comptel
LDAP_fax	No	+3589123456
LDAP_l	No	Helsinki
LDAP_postalAddress	No	Ruoholahdenkatu 99
LDAP_postalCode	No	00100
LDAP_postOfficeBox	No	PO123
LDAP_street	No	Ruoholahdenkatu 99
LDAP_telephoneNumber	No	+35897654321
LDAP_userPassword	No	Password
LDAP_vpnName	Yes	VPN_A
LDAP_vpnSiteName	Yes	Helsinki_1
LDAP_activationDate	No	30082001

Request Parameter	Mandatory	Example
LDAP_slaInformation	No	http://intra.sla.com/VPNCustomer/Comptel/
LDAP_contactPerson	No	Jukka Halonen

Response

Task execution successful

No specific LDAP response.

Task execution erroneous

A LDAP server's exception message of a LDAP macro server /macro set error message.

Modify

The LDAP parameter specific approach works fine with the modify method. Usually the modification of an LDAP entry consists of changing only few basic values. An example of the modify case could be e.g. that customer changes the CE device. The MDS/SAS must update only a few attributes. Otherwise this method is quite similar to create- method.

Response

Task execution successful

No specific LDAP response.

Task execution erroneous

A LDAP server's exception message of a LDAP macro server /macro set error message.

Delete

The delete functionality is straightforward. DN defines the entry that is to be deleted. There are no LDAP specific parameters. The mandatory DN parameter is the only required parameter.

Response

Task execution successful

No specific LDAP response.

Task execution erroneous

A LDAP server's exception message of a LDAP macro server /macro set error message.

Search

There is basically no difference to the template approach in the search functionality when using the LDAP specific parameter approach. The only difference is that the service request must contain all search parameters. All other points, such as the structure of the response, etc., are similar to the situation mentioned in the template approach.

Table 6-4: Request Parameters for Search

Request Parameter	Mandatory	Example
LDAP_DN	Yes	vpnName=VPN_A, ou=device, dc=GoldenConnection, dc=com
LDAP_SCOPE	Yes	SCOPE_BASE
LDAP_FILTER	Yes	(deviceId=VPN_A*)
LDAP_CONS_TIME	No	100
LDAP_CONS_NUMBER	No	200

Response

Task execution successful

Response is described in the previous chapter.

Task execution erroneous

A LDAP server's exception message of a LDAP macro server /macro set error message.

6.4 Technicalities

6.4.1 Logging Policy

Log entries are made to regular files. MDS/SAS specifies the file when the request execution starts. The log trace level is an integer value between 0 and 99. The bigger the value, the less important is the entry. The trace level is defines in the GRC file. The log and error files that Java Macro Server writes are depicted in Figure 6-3.

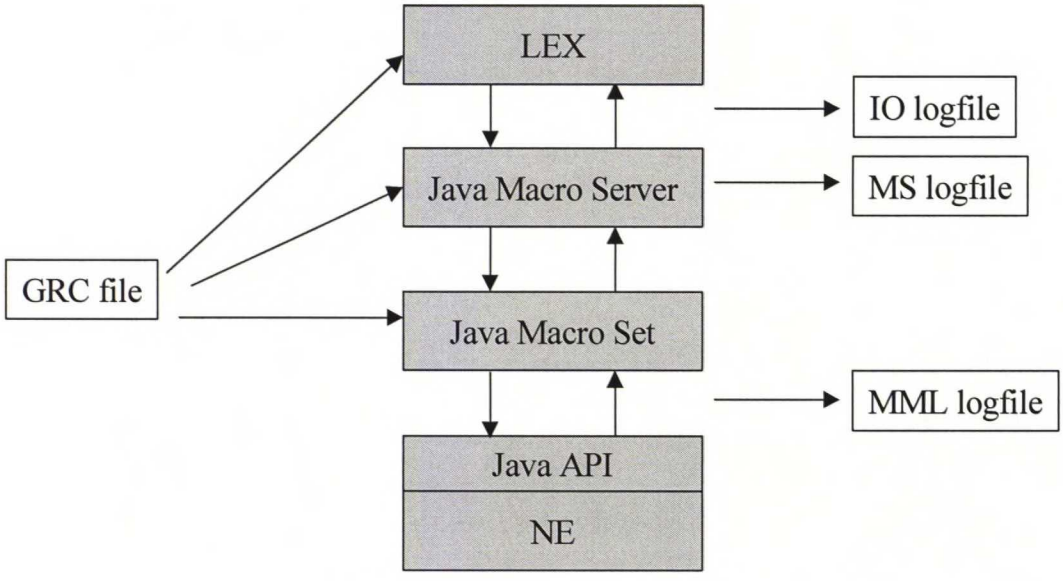


Figure 6-3: Log and GRC files

The IO log file contains communication between MDS/SAS LEX and Java Macro

Server. The file location is specified in the GRC file. Java Macro Server and LEX jointly writes logging information to the file. LEX writes the commands that it sends to the Java Macro Server and Java Macro Server writes responses to these commands.

The MML log file contains the information that is transferred between the Java Macro Set and the NE interface. Java Macro Server and Java Macro Set writes logging information to this file. The file location is specified in the GRC file.

When an error occurs in the Java Macro Server, it informs it by writing an entry into the error file. The file location is specified in the GRC file. Entry can be an exception from the LDAP directory or an error in the execution of the macro server. Also debug messages are printed to this error file. Macro Server writes always a debug message when a task is started. Macro set writes its own debug messages to the same file.

6.4.2 Template Grammar

Due to the differences of the LDAP operations two different template grammars are required. The first template grammar is for create, delete and modify operations. It allows multiple entries so that several entries can be created with one MDS/SAS request. This grammar follows the guidelines of LDIF notation with some changes. The second template grammar is for the search operation. This grammar is basically just a list of parameters that the search operation requires.

The two LDAP template grammars in extended BNF are described below. White spaces and comments of form /* [.\n]* */ are stripped during tokenising. Description of notation follows.

<>	non-terminal
{}	grouping
[ ]	optional
|	or

+	one or more
*	zero or more
TEXT	literal

<safe-char> includes any character which decimal value is less than or equal to 127 except NUL, LF and CR. <safe-init-char> includes any character which decimal value is less than or equal to 127 except NUL, LF, CR, SPACE, colon and less-than. These parameters follow the syntax definition in [RFC2849].

Template Grammar for Create, Delete and Modify Operations

```
<ldapprog>              ::= LDAPDATA( <entry>+ );
<entry>                 ::= ENTRY( <parameter> { ; <parameter> }* );
<parameter>             ::= <attributeType> <AttributeValuelist>
<attributeType>         ::= { [A-Za-z] | [0-9] }+
<attributeValuelist>    ::= <dn> | <attributeValue> { , <attributeValue> }*
<dn>                    ::= <safe_init_char> { <safe_char> }*
<safe_init_char>        ::= [0x01-09] | [0x0B-0C] | [0x0E-1F] | [0x21-39] | [0x3B] |
                            [0x3D-7F]
<safe_char>             ::= [0x01-09] | [0x0B-0C] | [0x0E-7F]
<attributeValue>        ::= <variable> | <string>
<variable>              ::= [<] <identifier> [>]
<identifier>            ::= { [A-Za-z] | [0-9] | _ | - }+
<string>                ::= "{ [^\n\t"\] | \[nt"\] }*"
```

Template Grammar for Search Operation

```
<ldapprog>              ::= LDAPDATA( <parameter> { ; <parameter>* } );
<parameter>             ::= BASE <baseValue> | SCOPE <scopeValue> |
                            FILTER <filterValue> | ATTRSONLY <attrsonlyValue>|
                            ATTRS <attrsValue> | TIMELIMIT <timelimitValue> |
                            SIZELIMIT <sizelimitValue>
<baseValue>             ::= <base> | <variable>
<base>                  ::=          <safe_init_char> { <safe_char> }*
```

<safe_init_char> ::= **[0x01-09] | [0x0B-0C] | [0x0E-1F] | [0x21-39] | [0x3B] | [0x3D-7F]**
<safe_char> ::= **[0x01-09] | [0x0B-0C] | [0x0E-7F]**
<scopeValue> ::= <scope> | <variable>
<scope> ::= **SCOPE_BASE | SCOPE_ONELEVEL | SCOPE_SUBTREE**
<filterValue> ::= <filter> | <variable>
<filter> ::= **{ [A-Za-z] | [0-9] | = | < | > | ~ | * | ! | , | (|) | & | | }+**
<attrsonlyValue> ::= <attrsonly> | <variable>
<attrsonly> ::= **TRUE | FALSE**
<attrsValue> ::= <attrs> | <variable>
<attrs> ::= **{ [A-Za-z] | [0-9] | = | , | (|) }+**
<timelimitValue> ::= <timelimit> | <variable>
<timelimit> ::= <number>
<sizelimitValue> ::= <sizelimit> | <variable>
<sizelimit> ::= <number>
<variable> ::= **[<]** <identifier> **[>]**
<number> ::= **[0-9]+**
<identifier> ::= **{ [A-Za-z] | [0-9] | _ | - }+**

6.4.3 GRC Parameters

A set of control parameters is used to control the behaviour of MDS/SAS. These parameters are maintained in a text file called GRC file. The location and the name of the GRC file is defined by the RCFILE- environment variable. The GRC file is divided into sections. Each section is identified by a name and can contain any number of variable definitions. Each variable is defined in its own line. Every MDS/SAS subsystem has its own section in addition to a common MDS/SAS section. Thus, both Java Macro Server and Java Macro Set have their own section for their GRC parameters.

Table 6-5: Java Macro Set GRC parameters

Section: ldap_macro_set	
GRC Parameter name	**Description**
prefix	Defines the prefix that is used when provisioning using LDAP specific parameters.
template_directory	Defines the location of LDAP templates when provisioning using the template approach.

Table 6-6: Java Macro Server GRC parameters

Section: java_macro_general	
GRC Parameter name	**Description**
display_new_services	Determines whether the new services of a subscriber are displayed in a response to create and modify operations. The values are: 0 = Do not show new services. 1 = Show new services.
display_old_services	Determines whether the services that a subscriber had before delete and modify operations are displayed in a response. The values are: 0 = Do not show old services. 1 = Show old services.
macro_config_dir	Specifies the directory where the macro server configuration files are located.
new_supp_coding	Determines the method of supplementary service coding. The possible values are: 1 = Use the new supplementary service coding, i.e. the action code 2 stands for withdrawal. 2 = Use the new supplementary service coding. Display also the withdrawn services in a response for a display request.
tracelevel	log screening level

7 Conclusions and Further Work

Virtual private network is one of the fundamental services that service providers must offer to their customers. VPN has several advantages over leased line, especially related to costs savings and scalability. As the technology evolves the importance of VPN becomes even more significant than it is today. MPLS seems to be the technology that will be used to build large-scale VPN solutions. MPLS based VPN solutions have clear benefits both for the service provider and for the customer. These are the reasons why MPLS based VPN is the first VPN solution that MDS/SAS is configured to support.

The tools used for service activation in MPLS based VPNs should be able to perform network management operations. MDS/SAS has features that enable many of the network management operations. However, one must remember that MDS/SAS is not a full-scale network management system. Should MDS/SAS have the capabilities to support service activation in MPLS based VPN, some design principles of MDS/SAS must be changed. E.g. MDS/SAS should get a network management information repository and more flexible GUI in order to support network management operations. Today MDS/SAS acts more or less as a highly sophisticated gateway to the network elements without maintaining any service related data.

LDAP is the most important interface that Comptel will support in IP world for many reasons. First of all every major software and hardware vendor have implemented LDAP interfaces to their products. It is also the key protocol for enabling DEN concept. DEN is an attractive concept from Comptel's point of view. MDS/SAS can be easily integrated to a DEN implementation. DEN basically enables MDS/SAS to act similarly in IP network as it does today in telecommunication networks.

The LDAP interface module implementation is quite straightforward. The two biggest problems are to find a solution that would enable a generic software module and to implement the search functionality. Template approach solves the first challenge.

However, changes must be made to the MDS/SAS core response functionality in order to manage large responses of search functionality. On the other hand, one can say that MDS/SAS is not designed to operate with large responses. If the customer requires bulk search functionality, it just has to use some other tool.

MDS/SAS is a good tool for provisioning different types of network elements that uses different types of interfaces. MDS/SAS can offer many advance services and features for the customer care systems. MDS/SAS should not be used for managing large data networks but in conjunction with network management systems. MDS/SAS can be used to connect network management systems to business systems or for provisioning network elements that are not supported by the network management systems. MDS/SAS works also fine with convergent networks, such as GPRS and UMTS, where it handles the provisioning to the telecommunication networks and possibly to some IP network elements, e.g. AAA server, DNS and service platforms.

References

[ATMForum] Technical Committee, "Traffic Management Specification Version 4.1", AF-TM-0121.000, March 1999

[Blac01] Black, U., "MPLS and Label Switching Networks", Prentice-Hall Inc., 2001

[Broa98] Broadband Publishing report, "Virtual Private Network and the Enterprise", 1998, http://www.atmreport.com/

[Chap99] Chappell, L., "Introduction to Cisco Router Configuration", Macmillan Technical Publishing, 1999

[Cisc99] Internetworking Technology Overview, "Directory-Enabled Networking", June 1999, http://www.cisco.com/

[Cisc01] Technical Service Description, "Intranet and Extranet Virtual Private Networking", Cisco Systems Inc, March 2001 http://www.cisco.com/

[Comp01] Comptel Technical Documentation, "MDS/SAS Functional Description", April 2001

[Data01] Datamonitor report, "Secure Remote Access Solutions: Profiting from the VPN opportunity", March 2001

[Davi00] Davie, B., Rechter, Y., "MPLS – Technology and Applications", Morgan Kaufmann, 2000

[DMTF] Distributed Management Task Force Inc., http://www.dmtf.org/, 2001

[Down98] Downes, K., Ford, M., Kim Lew, H., Spanier, S., Stevenson, T., "Internetworking Technologies Handbook", Macmillan Technical Publishing, 2nd Edition, 1998

[FRForum] Frame Relay Forum, "Basic Guide to Frame Relay Networking", 1998, http://www.frforum.com/basicsguide.html

[Hall96] Halsall, F., "Data Communications, Computer Networks and Open Systems", Addison-Wesley, 4th Edition, 1996

[Huuh99] Huuhtanen, J., "Palveluprovisiointijärjestelmän sisäisen palvelukuvauksen määrittely ja toteutuksen suunnittelu", Master's Thesis, Lappeenranta University of Technology, Information Technology, 1999

[ITU.211] ITU-T Recommendation I.211, "B-ISDN service aspects ", March 1993

[Java01] Comptel Technical Documentation, "MDS/SAS Java Macro Server - Technical Description", 2001

[John98] Johner, F., Brown, L., Hinner, F., Reis, W., Westman, J., "Understanding LDAP", IBM Corporation, International technical Support Organization, 1st Edition, June 1998

[Judd98] Judd, S., Strassner,. J. "Directory-enabled Networks – Information Model and Base Schema", February 1998

[Korp95] Korpela, P., "Subscriber Administration of telecommunications network", Master's Thesis, Helsinki University of Technology, Faculty of information technology, September 1995

[Kosi98] Kosiur, D., "Building and Managing Virtual Private Networks", Wiley Computer Publishing, 1998

[Moto00] Motoszko, J., "IP VPN – Service for the new millennium", Presentation, Cisco Forum, March 2000

[MPLS01] Cisco Documentation, "Configuring Multiprotocol Label Switching", Jan 2001, http://www.cisco.com/

[Pyyh99] Pyyhtiä, M., "Palvelunlaatu Heterogeenisissä IP-verkoissa", Master's Thesis, Helsinki University of Technology, Electrical and Communications Engineering, 1999

[RFC1226] Kantor, B. "Internet Protocol Encapsulation of AX.25 Frames", May 1991, http://www.ietf.org/rfc/rfc1226.txt

[RFC1234] Provan, D., "Tunneling IPX Traffic through IP Networks", June 1991, http://www.ietf.org/rfc/rfc1234.txt

[RFC1701] Hanks, S., Li, T., Farinacci, D., Traina, P., "Generic Routing Encapsulation (GRE)", October 1994, http://www.ietf.org/rfc/rfc1701.txt

[RFC1823] Howes, T., Smith, M., "The LDAP Application Program Interface", August 1995, http://www.ietf.org/rfc/rfc1823.txt

[RFC1853] Simpson, W., "IP in IP Tunneling", October 1995, http://www.ietf.org/rfc/rfc1853.txt

[RFC2251] Wahl, M., Howes, T., Kille, S., "Lightweight Directory Access Protocol (v3)", December 1997, http://www.ietf.org/rfc/rfc2251.txt

[RFC2252] Wahl, M., Coulbeck, Howes, T., A., Kille, S., "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions", December 1997, http://www.ietf.org/rfc/rfc2252.txt

[RFC2254] Howes, T., "The String Representation of LDAP Search Filters", December 1997, http://www.ietf.org/rfc/rfc2254.txt

[RFC2283] Bates, T., Chandra, R., Katz, D., Rekhter, Y., "Multiprotocol Extensions for BGP-4", February 1998, http://www.ietf.org/rfc/rfc2283.txt

[RFC2401] Kent, S., Atkinson, R., "Security Architecture for the Internet Protocol", November 1998, http://www.ietf.org/rfc/rfc2401.txt

[RFC2547] Rosen, E., Rekhter, Y., "BGP/MPLS VPNs", March 1999, http://www.ietf.org/rfc/rfc2547.txt

[RFC2661] Townsley, W., Valencia, A., Rubens, a., Pall, G., Zorn, G., Palter, B., "Layer Two Tunneling Protocol "L2TP" ", August 1999, http://www.ietf.org/rfc/rfc2661.txt

[RFC2702] Awduche, A., Malcolm, J., Agogbua, J., O'Dell, M., McManus, J., "Requirements for Traffic Engineering Over MPLS", September 1999 http://www.ietf.org/rfc/rfc2702.txt

[RFC2764] Gleeson, B., Lin, A., Heinanen, J., Armitage, G., Malis, A., "A Framework for IP Based Virtual Private Networks", February 2000, http://www.ietf.org/rfc/rfc2764.txt

[RFC2828] Shirey, R., "Internet Security Glossary", May 2000, http://www.ietf.org/rfc/rfc2828.txt

[RFC2829] Wahl, M., Alvestrand, A., Hodges, J., Morgan, R., "Authentication Methods

for LDAP ", May 2000, http://www.ietf.org/rfc/rfc2829.txt

[RFC2849] Good, G., "The LDAP Data Interchange Format (LDIF) - Technical Specification", June 2000, http://www.ietf.org/rfc/rfc2849.txt

[RFC2891] Howes, T., Wahl, M., Anantha, A., "LDAP Control Extension for Server Side Sorting of Search Results", August 2000, http://www.ietf.org/rfc/rfc2891.txt

[RFC3031] Rosen, E., Viswanathan, A., Callon, R., "Multiprotocol Label Switching Architecture", January 2001, http://www.ietf.org/rfc/rfc3031.txt

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., Conta, A., "MPLS Label Stack Encoding ", January 2001, http://www.ietf.org/rfc/rfc3033.txt

[RFC3036] Andersson, L., Doolan, P., Feldman, N., Fredette, A., Thomas, B., "LDP Specification", January 2001, http://www.ietf.org/rfc/rfc3036.txt

[RFC3062] Zeilenga, K., "LDAP Password Modify Extended Operation ", February 2001, http://www.ietf.org/rfc/rfc3062.txt

[RPROT] Cisco Documentation, "IP Routing Protocols", 2001, http://www.cisco.com/

[Wilc99] Wilcox, M., "Implementing LDAP", Wrox Press Ltd, 1999

Appendix 1: Create Template

```
/*********************************************************************
PROJECT: MDS/SAS/LDAP Macro Server
TemplateName:          Create_VPN_customer.ldap
SAS version:           3.4.x or higher
Macro server version: 1.x
$Source: /SAS/macro_server/dippa/templates/Create_VPN_customer.ldap $
Author: Jukka Halonen
$Modified$
@(#)$Id:  Create_VPN_customer.ldap /main/1 2001/09/31 17:00:00 jhalonen
CHECKEDOUT $
Copyright (c) Comptel Plc 2001
*********************************************************************/
LDAPDATA(
/* First entry is used for adding the customer information */
ENTRY(
      dn <DN_CUSTOMER>;
      objectClass "top,organizationalUnit,organizationalUnitCust";
      ou <OU>;
      description "Not used";
      fax <FAX>;
      internationalIsdnNumber "Not used";
      l <LOCATION>;
      postalAddress <POSTALADDRESS>:
      postalCode <POSTALCODE>
      postOfficeBox <POSTOFFICEBOX>;
      seeAlso "Not used";
      st "Not used";
      street <STREET>;
      telephoneNumber <TN>;
      userPassword <PASSWORD>
      vpnName <VPN_NAME>;
      vpnSiteName <VPN_SITE_NAME>;
      vpnStatus "Not used";
      activationDate <ACTIVATION_DATE>;
      deactivationDate "To be used with new CCS";
      slaInformation <SLA_URL>;
```

```
        contactPerson <CONTACT_PERSON>
);
/* Second entry is used for adding the device information */
ENTRY(
        dn <DN_DEVICE>;
        objectClass "top,device,deviceCust";
        cn <CE_NAME>;
        l <LOCATION>;
        ou <OU>;
        owner "Not used";
        seeAlso "Not used";
        serialNumber <SERIALNUMBER>;
        deviceId <DEVICEID>;
        peerId <PEERID>;
        vpnName <VPN_NAME>;
        vpnSiteName <VPN_SITE_NAME>;
        ipAddress <IPADDRESS>;
        deviceVendor <VENDOR>;
        deviceModel <MODEL>;
        deviceOS <OS>;
        StatusInfo "Not used";
        macAddress <MACADDRESS>
);
/* Third entry is used for adding the connection information */
ENTRY(
        dn <DN_CONNECTION>;
        objectClass "top,connectionCust";
        subnetNumber <SUBNET>;
        subnetMask <SUBNETMASK>;
        vpnName <VPN_NAME>;
        vpnSiteName <VPN_SITE_NAME>;
        addressType "IPv4";
        protocolType "IPv4";
        lanType "Ethernet";
        routingProtocol "BGP"
);); /* EOF */
```

Appendix 2: Search Template

```
/*************************************************************************
PROJECT: MDS/SAS/LDAP Macro Server
TemplateName:          Search_VPN_customer.ldap
SAS version:           3.4.x or higher
Macro server version: 1.x
$Source: /SAS/macro_server/dippa/templates/Search_VPN_customer.ldap $
Author: Jukka Halonen
$Modified$
@(#)$Id:  Create_VPN_customer.ldap /main/1 2001/09/31 17:00:00 jhalonen
CHECKEDOUT $
Copyright (c) Comptel Plc 2001
*************************************************************************/
LDAPDATA(
BASE <SEARCHBASE>;
FILTER <FILTER>;
SCOPE SCOPE_SUBTREE;
ATTRONLY "FALSE";
ATTRS "cn,mail,telephoneNumber";
TIMELIMIT 100;
SIZELIMIT 12
);
```

Appendix 3: Router Configuration

This appendix describes the configurations that PE and CE devices require in order to activate VPN routes between the sites of VPN_A. The whole router configuration files are omitted due to their length. On the other hand these are the configurations that MDS/SAS must be able to perform in order to activate the VPN. VPN_A is set-up between Helsinki_1, Kerava_1, Kerava_2 and Tampere. Other sites do not have an access to the VPN_A.

7.1 CE devices

As mentioned earlier the only required configuration on CE devices is the activation of the BGP session. All the configurations on CE devices CE-A, CE-B, CE-G and CE-H are similar. Below is the configuration information that must be added to the CE-G (Helsinki_1) device. The CE devices require information only about the routing-peer (i.e. PE-device) thus the number of VPN sites in VPN_A does not have an effect on the CE device configuration.

7.1.1 CE-G (Helsinki_1)

```
ROUTER BGP 107
NETWORK 20.7.0.0 MASK 255.255.255.0
NETWORK 20.6.6.6 MASK 255.255.255.255
NEIGHBOR 20.7.0.1 REMOTE-AS 100
```

7.2 PE devices

PE devices require a lot more configuration than CE devices. Here is the configuration information that MDS/SAS must be able to add on the PE_F (Helsinki) router in order to activate VPN_A on site Helsinki_1. Again the configurations in the PE devices are quite

similar so only configuration information of the PE-F router is presented here. Only those PE devices that are part of the VPN_A must be introduced in the PE device provisioning.

7.2.1 PE-F (Helsinki)

```
IP VRF VPN_A
RD 109:1
ROUTE-TARGET EXPORT 109:1
ROUTE-TARGET IMPORT 109:1
IP CEF
!
INTERFACE LOOPBACK0
IP ADDRESS 10.0.0.15 255.255.255.255
!
INTERFACE FASTETHERNET0/0
IP VRF FORWARDING VPN_A
IP ADDRESS 20.7.0.1 255.255.255.0
!
INTERFACE ATM2/0.1 TAG-SWITCHING
IP ADDRESS 10.35.0.1 255.255.255.0
NO IP DIRECTED-BROADCAST
TAG-SWITCHING IP
!
AUTONOMOUS-SYSTEM 100
!
ROUTER OSPF 1
REDISTRIBUTE BGP 100
NETWORK 10.0.0.0 0.255.255.255 AREA 0
!
ROUTER BGP 100
NO SYNCHRONIZATION
NEIGHBOR 10.0.0.10 REMOTE-AS 100
NEIGHBOR 10.0.0.10 UPDATE-SOURCE LOOPBACK0
!
ROUTER BGP 100
NO SYNCHRONIZATION
NEIGHBOR 10.0.0.16 REMOTE-AS 100
NEIGHBOR 10.0.0.16 UPDATE-SOURCE LOOPBACK0
!
ADDRESS-FAMILY IPV4 VRF VPN_A
NEIGHBOR 20.7.0.2 REMOTE-AS 101
NEIGHBOR 20.7.0.2 ACTIVATE
NEIGHBOR 20.7.0.2 REMOTE-AS 102
NEIGHBOR 20.7.0.2 ACTIVATE
NEIGHBOR 20.7.0.2 REMOTE-AS 108
NEIGHBOR 20.7.0.2 ACTIVATE
```

```
NO AUTO-SUMMARY
NO SYNCHRONIZATION
EXIT-ADDRESS-FAMILY
!
ADDRESS-FAMILY VPNV4
NEIGHBOR 10.0.0.10 ACTIVATE
NEIGHBOR 10.0.0.10 SEND-COMMUNITY EXTENDED
NEIGHBOR 10.0.0.16 ACTIVATE
NEIGHBOR 10.0.0.16 SEND-COMMUNITY EXTENDED
EXIT-ADDRESS-FAMILY
!
```